

Grado en Ingeniería Informática

2019-2020

Trabajo Fin de Grado

Automatización y control de persianas eléctricas mediante microcontroladores

David Silveira Amaro

Tutor

Alberto Cascajo García

Colmenarejo, 2020



Esta obra se encuentra sujeta a la licencia Creative Commons
Reconocimiento – No Comercial – Sin Obra Derivada

RESUMEN

La tecnología en el mundo actual avanza a pasos agigantados, en donde cada vez más se busca una automatización y conexión con la nube de cualquier tipo de elemento. El control y automatización de una vivienda mediante la tecnología se conoce como domótica, teniendo como principal objetivo la gestión eficiente de los recursos maximizando el confort de los individuos. Este proyecto surge de la idea de desarrollar un componente domótico con el que controlar y automatizar las persianas de una vivienda, ofreciendo ventajas frente a las alternativas existentes en el mercado.

En la actualidad, la palabra domótica viene, inevitablemente, acompañada de los teléfonos inteligentes ya que no hay nada más cómodo y fácil para el usuario que tener el control en la palma de la mano. Dado que el proyecto consiste en la creación de un sistema domótico completamente funcional, se llevará a cabo el desarrollo de un componente capaz de controlar un motor de persiana y de una aplicación móvil que permitirá gestionar todas las persianas domotizadas.

En primer lugar, se diseña y desarrolla el dispositivo que permitirá el control de un motor de persiana genérico. Para ello se utiliza un microcontrolador (ESP32), el cual permite establecer una conexión Wi-Fi que será necesaria para la comunicación con la aplicación móvil. El dispositivo desarrollado contiene botones que permiten un control simple de la persiana y utilizará la conexión Wi-Fi para ser controlado y configurado desde el dispositivo móvil. En segundo lugar, se desarrollará una aplicación móvil para el sistema operativo Android con el cual gestionar de forma sencilla y eficiente las persianas conectadas a la red Wi-Fi.

Tras la implementación del proyecto se han obtenido los resultados con varios dispositivos de control conectados simultáneamente, y con uno de ellos funcionando en una persiana real. Los resultados aportados por el proyecto cumplen los objetivos en cuanto a funcionamiento y costes, permitiendo la domotización completa de las persianas de una casa de forma inalámbrica y por mucho menos dinero que con los productos que ofrece el mercado.

Palabras clave

Microcontrolador, Automatización, Domótica, Aplicación móvil, Persiana.

ABSTRACT

Technology, in today's world, is advancing by leaps and bounds, where more and more people are looking for automation and connection to the cloud of any type of element. The control and automation of a house through technology is known as domotics, having as main objective the efficient management of resources maximizing the comfort of people. This project arises from the idea of developing a domotic component with which to control and automate the blinds of a home, offering advantages over existing alternatives on the market.

Nowadays, the home automation is inevitably accompanied by smartphones, as there is nothing more comfortable and easy for the user than having control in their phone. The project consists of the creation of a fully functional home automation system, what it entails the development of a component capable of controlling a blind motor and the mobile application that will manage all the domotic blinds.

The first step is the desing and development of the device that will allow the control of a generic blind motor. For this purpose, a microcontroller (ESP32) is used, which allows the establishment of a Wi-Fi connection, that will be necessary for communication with the mobile application. The developed device contains buttons that allow simple control of the blind, and will use the Wi-Fi connection to be controlled and configured from the smartphone. The mobile application will be developed for the Android operating system allowing an easily and efficiently manage of the blinds connected to the Wi-Fi network.

After the implementation of the project, the results have been tested with several control devices connected simultaneously and with one of them operating in a real blind. The results provided by the project reach the goals in terms of functionality and costs, allowing the complete wireless domotization of the blinds of a house and for much less money than with the products offered in the market.

Keywords

Microcontroller, Automation, Domotics, Mobile application, Blind.

AGRADECIMIENTOS

Me gustaría agradecer toda la ayuda y compañía que me han dado mis amigos durante esta etapa de mi vida.

A mi tutor, Alberto, por apoyarme y aconsejarme en este proyecto.

Por último, a mi familia, que siempre me han ayudado y apoyado con todo.

ÍNDICE DE CONTENIDOS

1.	INTRODUCCIÓN	1
1.1.	MOTIVACIÓN DEL TRABAJO.....	1
1.2.	OBJETIVOS.	1
1.3.	MARCO REGULADOR.	2
1.4.	ESTRUCTURA DEL DOCUMENTO	3
2.	ESTADO DEL ARTE	4
2.1.	SITUACIÓN ACTUAL.	4
2.1.1.	DOMÓTICA EN LA ACTUALIDAD.....	4
2.1.2.	INVESTIGACIONES Y PROYECTOS PREVIOS SOBRE DOMÓTICA.	5
2.1.3.	ACTUALIDAD DEL MERCADO.....	6
2.2.	DISEÑO DE SOLUCIONES.	7
3.	ANÁLISIS	8
3.1.	DESCRIPCIÓN DEL PROBLEMA.	8
3.1.1.	PERSPECTIVA DEL PRODUCTO.	8
3.1.2.	ALCANCE DEL PROYECTO.	8
3.2.	REQUISITOS.....	9
3.2.1.	REQUISITOS FUNCIONALES.....	9
3.2.2.	MATRIZ DE TRAZABILIDAD.	19
3.2.3.	REQUISITOS NO FUNCIONALES.....	20
3.3.	CASOS DE USO.	20
3.3.1.	DIAGRAMA DE CASOS DE USO.	20
3.3.2.	CASOS DE USO DE ALTO NIVEL.	21
4.	DISEÑO	25
4.1.	DESCRIPCIÓN DE LA ARQUITECTURA.	25
4.2.	DIAGRAMA DE CLASES.	25
4.3.	DIAGRAMA DE COMPONENTES.	26
4.4.	DESCRIPCIÓN DEL ESQUEMA DE COMUNICACIONES.....	26
4.5.	DISEÑO DEL DCP.	27
4.5.1.	MICROCONTROLADOR.....	27
4.5.2.	MÓDULO RELÉ.	30
4.5.3.	MÓDULO RTC.	32
4.5.4.	CONSUMO DE CORRIENTE DEL DCP.....	33
4.6.	DISEÑO DE LA APLICACIÓN.....	34
4.6.1.	SISTEMA OPERATIVO Y ENTORNO DE DESARROLLO.	34
4.6.2.	PROTOTIPO DE INTERFAZ DE USUARIO.....	35
5.	IMPLEMENTACIÓN.....	40
5.1.	IMPLEMENTACIÓN DEL DCP.	40
5.1.1.	ESQUEMA ELÉCTRICO.	40
5.1.2.	CODIFICACIÓN.	42
5.2.	IMPLEMENTACIÓN DE LA APLICACIÓN.....	47
5.2.1.	CODIFICACIÓN.	47
5.3.	PRUEBAS.	49
5.3.1.	PRUEBAS DEL DCP.	49
5.3.2.	PRUEBAS DE LA APLICACIÓN.....	50

5.3.3.	PRUEBAS CONJUNTAS.....	50
6.	RESULTADOS	51
6.1.	RESULTADOS DEL DCP.	51
6.2.	RESULTADOS DE LA APLICACIÓN.....	53
6.3.	RESULTADO GENERAL DEL PROYECTO.....	57
7.	PLANIFICACIÓN Y COSTES	59
7.1.	PLANIFICACIÓN INICIAL.	59
7.1.1.	DIAGRAMA DE GANTT.	59
7.2.	ESTIMACIÓN DE COSTES.	60
7.3.	PRESUPUESTO.....	61
7.4.	DEDICACIÓN REAL.....	62
7.5.	COSTES REALES.	64
7.6.	IMPACTO SOCIOECONÓMICO.	65
8.	CONCLUSIONES Y TRABAJOS FUTUROS.....	66
8.1.	CONCLUSIONES.	66
8.2.	LÍNEAS FUTURAS DE TRABAJO.....	67
9.	ENGLISH COMPETENCE	68
9.1.	INTRODUCTION.	68
9.2.	OBJECTIVES.	68
9.3.	MARKET RESEARCH.....	69
9.4.	PROJECT SCOPE.....	69
9.5.	DESIGN.	70
9.5.1.	WI-FI COMMUNICATIONS.	70
9.5.2.	DCP.	70
9.5.3.	APP.	71
9.6.	RESULTS.	72
9.6.1.	DCP RESULTS.	72
9.6.2.	APP RESULTS.	73
9.6.3.	GENERAL RESULTS OF THE PROJECT.....	74
9.7.	PLANNING AND COSTS.	75
9.8.	CONCLUSIONS.....	76
9.9.	FUTURE WORK.	77
	BIBLIOGRAFÍA	78
	ANEXO A: MANUAL DE USO.....	80

ÍNDICE DE FIGURAS

FIG. 2.1. CODIFICACIÓN DE BITS EN EL PROTOCOLO X10	5
FIG. 2.2. CONEXIONES DEL MOTOR DE PERSIANA.....	7
FIG. 3.1. DIAGRAMA DE CASOS DE USO	21
FIG. 4.1. DIAGRAMA DE CLASES	25
FIG. 4.2. DIAGRAMA DE COMPONENTES	26
FIG. 4.3. MICROCONTROLADOR ESP32	29
FIG. 4.4. MÓDULO RELÉ DE DOS CANALES	31
FIG. 4.5. MÓDULO RTC DS3231.....	32
FIG. 4.6. MOCK-UP PANTALLA PRINCIPAL	35
FIG. 4.7. MOCK-UP PANTALLA DE CONTROL	36
FIG. 4.8. MOCK-UP PANTALLA EDICIÓN DE PROGRAMACIÓN HORARIA.....	37
FIG. 4.9. MOCK-UP PANTALLA DE CREACIÓN DE GRUPO	38
FIG. 4.10. MOCK-UP PANTALLA DE SINCRONIZACIÓN DE DESPLAZAMIENTO.....	39
FIG. 5.1. DISPOSICIÓN DE LOS PINES DEL ESP32.....	41
FIG. 5.2. DISEÑO DEL CIRCUITO DEL DCP	42
FIG. 5.3. DISTRIBUCIÓN DE DISPOSITIVOS COMPATIBLES POR API DE ANDROID	47
FIG. 5.4. CAPTURA DE PANTALLA DEL MONITOR SERIAL.....	49
FIG. 6.1. CIRCUITO DEL DCP EN UNA PROTOBOARD	51
FIG. 6.2. CONEXIÓN ELÉCTRICA CON EL MÓDULO RELÉ	52
FIG. 6.3. APLICACIÓN - PANTALLA PRINCIPAL (DCP)	53
FIG. 6.4. APLICACIÓN – AÑADIR UN DCP A LA RED WI-FI.....	54
FIG. 6.5. APLICACIÓN – CONFIGURACIÓN DE UN DCP.....	55
FIG. 6.6. APLICACIÓN – CREAR UN NUEVO GRUPO	56
FIG. 6.7. APLICACIÓN – CONFIGURACIÓN DE UN GRUPO	57
FIG. 7.1. DIAGRAMA DE GANTT DE LA PLANIFICACIÓN INICIAL.....	59
FIG. A.1. CONEXIONES BAJADA Y SUBIDA	80
FIG. A.2. CONEXIONES FASE Y NEUTRO	81
FIG. A.3. BOTONES DEL DISPOSITIVO DE CONTROL	82
FIG. A.4. INDICADORES DE ESTADO LED.....	83
FIG. A.5. OPCIONES DE CONTROL DE UNA PERSIANA	85
FIG. A.6. OPCIONES DE CONTROL DE UN GRUPO	86

ÍNDICE DE TABLAS

TABLA 3.1. PLANTILLA DE REQUISITOS FUNCIONALES	9
TABLA 3.2. REQUISITO FUNCIONAL RF-DCP-01.....	10
TABLA 3.3. REQUISITO FUNCIONAL RF-DCP-02.....	10
TABLA 3.4. REQUISITO FUNCIONAL RF-DCP-03.....	11
TABLA 3.5. REQUISITO FUNCIONAL RF-DCP-04.....	11
TABLA 3.6. REQUISITO FUNCIONAL RF-DCP-05.....	12
TABLA 3.7. REQUISITO FUNCIONAL RF-DCP-06.....	12
TABLA 3.8. REQUISITO FUNCIONAL RF-DCP-07.....	13
TABLA 3.9. REQUISITO FUNCIONAL RF-DCP-08.....	13
TABLA 3.10. REQUISITO FUNCIONAL RF-DCP-09.....	14
TABLA 3.11. REQUISITO FUNCIONAL RF-DCP-10.....	14
TABLA 3.12. REQUISITO FUNCIONAL RF-DCP-11.....	15
TABLA 3.13. REQUISITO FUNCIONAL RF-DCP-12.....	15
TABLA 3.14. REQUISITO FUNCIONAL RF-APP-01.....	15
TABLA 3.15. REQUISITO FUNCIONAL RF-APP-02.....	16
TABLA 3.16. REQUISITO FUNCIONAL RF-APP-03.....	16
TABLA 3.17. REQUISITO FUNCIONAL RF-APP-04.....	16
TABLA 3.18. REQUISITO FUNCIONAL RF-APP-05.....	17
TABLA 3.19. REQUISITO FUNCIONAL RF-APP-06.....	17
TABLA 3.20. REQUISITO FUNCIONAL RF-APP-07.....	17
TABLA 3.21. REQUISITO FUNCIONAL RF-APP-08.....	18
TABLA 3.22. REQUISITO FUNCIONAL RF-APP-10.....	18
TABLA 3.23. REQUISITO FUNCIONAL RF-APP-11.....	18
TABLA 3.24. REQUISITO FUNCIONAL RF-APP-12.....	19
TABLA 3.25. MATRIZ DE TRAZABILIDAD	19
TABLA 3.26. REQUISITOS NO FUNCIONALES.....	20
TABLA 3.27. CASO DE USO CU-01.....	21
TABLA 3.28. CASO DE USO CU-02.....	22
TABLA 3.29. CASO DE USO CU-03.....	22
TABLA 3.30. CASO DE USO CU-04.....	22
TABLA 3.31. CASO DE USO CU-05.....	23
TABLA 3.32. CASO DE USO CU-06.....	23
TABLA 3.33. CASO DE USO CU-07.....	23
TABLA 3.34. CASO DE USO CU-08.....	24
TABLA 3.35. CASO DE USO CU-09.....	24
TABLA 4.1. ESPECIFICACIONES ESP32	28
TABLA 4.2. ESPECIFICACIONES NODEMCU ESP8266.....	29
TABLA 4.3. ESPECIFICACIONES MÓDULO RELÉ	31
TABLA 4.4. ESPECIFICACIONES MÓDULO DS3231.....	32
TABLA 4.5. CONSUMO DE CORRIENTE DEL DCP.....	33
TABLA 5.1. ÓRDENES PERMITIDAS DEL DCP.....	44
TABLA 6.1. COSTE UNITARIO DEL DCP.....	58
TABLA 7.1. PLANIFICACIÓN INICIAL.	60
TABLA 7.2. ESTIMACIÓN DE COSTES – HARDWARE.	60
TABLA 7.3. ESTIMACIÓN DE COSTES – MENSUAL.....	60
TABLA 7.4. ESTIMACIÓN DE COSTES – TOTAL	61

TABLA 7.5. PRESUPUESTO.....	62
TABLA 7.6. DEDICACIÓN REAL.....	63
TABLA 7.7. COSTE REAL – HARDWARE.	64
TABLA 7.8. COSTE REAL – TOTAL.....	65

ACRÓNIMOS

AC	Alternating Current - Corriente alterna.
DC	Direct Current – Corriente directa.
DCP	Dispositivo de Control de Persianas.
IDE	Integrated Development Enviroment – Entorno de desarrollo integrado.
IoT	Internet of Things – Internet de las cosas.
IP	Internet Protocol – Protocolo de internet.
RTC	Real Time Clock – Reloj de tiempo real.
TCP	Transmission Control Protocol – Protocolo de control de transmisión.
UDP	User Datagram Protocol – Protocolo de datagramas de usuario.
Wi-Fi	Wireless Fidelity – Fidelidad inalámbrica.
WLAN	Wireless Local Area Network – Red inalámbrica de área local.

1. INTRODUCCIÓN

1.1. Motivación del trabajo.

La creciente tendencia de domotización del hogar combinada y propulsada por la aparición de los teléfonos inteligentes, y más recientemente, por los asistentes virtuales, ha disparado la necesidad de controlar y automatizar gran variedad de elementos de la vivienda. Las evoluciones tecnológicas producidas esta década han abierto un gran repertorio de posibilidades permitiendo la domotización y conectividad de muchos elementos como luces, termostatos y enchufes inteligentes.

Este trabajo nace de la idea personal de realizar un proyecto de domótica completamente funcional y sobre el cual aplicar y plasmar los conocimientos adquiridos. En este proyecto se busca diseñar y desarrollar un componente hardware capaz de acoplarse a un sistema de persiana motorizada, así como el software que permita el control y programación de la persiana desde un dispositivo móvil. Se pretende conseguir un dispositivo apto para el público general y con unas capacidades y características similares o superiores a las que ofrece el mercado actualmente. Además, se busca crear un dispositivo de bajo coste, abaratando el precio global que puede suponer la domotización de una vivienda, abriendo así esta posibilidad a un mayor número de personas.

Si bien este proyecto está basado y orientado a la domotización de persianas eléctricas, el producto resultante será perfectamente utilizable en otros elementos que sigan el mismo esquema de motor eléctrico, como podría ser una puerta de garaje o un toldo eléctrico.

1.2. Objetivos.

La finalización de este trabajo está condicionada a la consecución de tres objetivos principales:

- **O1:** El primer objetivo del proyecto es el diseño y creación de un dispositivo hardware basado en un microcontrolador, acoplando los componentes necesarios que permitan el control del motor de una persiana eléctrica. Además, el dispositivo de control de persianas (en adelante abreviado como DCP) debe permitir una conexión y comunicación vía Wi-Fi.
- **O2:** El segundo objetivo consiste en la implementación software del DCP, de modo que atienda peticiones vía Wi-Fi para controlar la persiana motorizada. También debe ser capaz de almacenar una programación horaria que permita a la

persiana realizar desplazamientos de manera autónoma en los días y horas establecidas por el usuario.

- **O3:** Como tercer objetivo, el desarrollo de una aplicación para dispositivos móviles capaz de gestionar, controlar y mostrar el estado de las persianas conectadas mediante el sistema domótico de la red Wi-Fi. Además, debe permitir configurar la programación de aperturas y cierres automatizando los movimientos de las persianas.

1.3. Marco regulador.

En esta sección se realiza un repaso de la legislación española y europea que afectan directamente al desarrollo de este TFG. El objetivo es, en caso de querer publicar la aplicación que se ha desarrollado, tener claras las leyes que son de cumplimiento obligatorio.

La primera referencia legal es la Constitución Española. En ella se puede observar el artículo 18, que dice textualmente *“Se garantiza el derecho al honor, a la intimidad personal y familiar y a la propia imagen. El domicilio es inviolable. Ninguna entrada o registro podrá hacerse en él sin el consentimiento del titular o resolución judicial, salvo en caso de flagrante delito”*. Esto quiere decir que la aplicación debe garantizar el secreto de las comunicaciones, y preservar el honor e intimidad de las personas. [1]

La segunda referencia es la Ley de Propiedad Intelectual (LPI). Sobre este tema hay diversas discusiones, ya que las aplicaciones informáticas (*software*) no tienen un método claro para llevar a cabo este objetivo. Las dos vertientes más importantes son: considerar los derechos de autor, o la utilización de patentes industriales y semejantes. Los derechos de autor protegen en cierta medida el software, pero tiene ciertas limitaciones y no es necesario un proceso de aceptación. Sin embargo, el sistema que se propone en este trabajo sí está sujeto a patentabilidad, ya que incluye elementos hardware y tienen una aplicación, si bien es reducida, a nivel industrial. Esto quiere decir que el sistema se podría patentar en su conjunto, tras pasar un complejo proceso de admisión, y garantizaría la explotación de la idea y del dispositivo. [2]

La tercera referencia legal a tener en cuenta es la Ley General de la Comunicación Audiovisual, que pretende regular la normativa vigente para que todos los medios de comunicación (hasta hace unos años radio y televisión, y ahora incluyendo diversos canales como las redes sociales o las páginas web) tengan las normas básicas sobre los derechos de contenidos, lo que influye en este trabajo desde el momento en el que se quiera publicar. [3]

Finalmente, hay que tener en cuenta la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD). Hoy en día, incluso los datos que aparecen en las tarjetas de visita que dejan algunas empresas se consideran de carácter personal. Por ello, es importante conocer los límites legales que debemos cumplir para poder llevar a cabo un

proyecto que incluye tratar con datos de personas físicas. Por ejemplo, informar al usuario de los datos que se le van a solicitar y se van a guardar, cómo se van a guardar y a utilizar dichos datos, qué otros componentes de software serán utilizados para proveer la seguridad, y un largo etc. En este proyecto no será necesario el uso de otros componentes externos, pero sí se deberá informar al usuario acerca de los datos para la inicialización del sistema, así como de la manera en la que se almacenarán los datos introducidos. [4]

1.4. Estructura del documento

El principal propósito del documento es describir y formalizar la funcionalidad que presenta el proyecto, así como detallar y argumentar las decisiones de diseño, problemas encontrados y soluciones aplicadas.

El documento se organiza de la forma que se indica a continuación: en la sección “Estado del Arte” se realiza un análisis del contexto actual de la domótica, investigando las soluciones que ofrece el mercado al problema en cuestión. Se investigan estudios y proyectos relacionados con las que aprender diferentes enfoques para aplicar soluciones a problemas relacionados con la domótica. En la sección de “Análisis” se muestra un estudio acerca del problema que se quiere resolver y el alcance del proyecto que se quiere desarrollar, incluyendo el análisis de requisitos que definirá el producto final. En “Diseño” se ofrece una visión de las alternativas que se han estudiado a la hora de llevar a cabo este proyecto, identificando qué soluciones pueden ser mejores o peores, y por qué se ha optado por una de ellas. En el capítulo de “Implementación” se muestra, con detalle, lo que se ha desarrollado para la consecución de los objetivos, tanto a nivel de software, como de hardware. En el capítulo de “Resultados” se muestra, de forma gráfica, el producto final que se ha desarrollado con este proyecto, que incluye una parte software para la aplicación de control, así como la parte hardware que manejará los dispositivos del hogar. Para finalizar, se detallan la planificación realizada previamente, así como la real, el presupuesto en base a las horas dedicadas y los costes estimados, y un capítulo donde se describen las conclusiones y las líneas de trabajo que se pueden seguir tras la conclusión de este trabajo.

En la última sección del documento se recogen los aspectos más relevantes y significativos del proyecto redactados en inglés.

2. ESTADO DEL ARTE

La domótica surgió como una oportunidad tecnológica que permitía mejorar la calidad de vida aportando confort y bienestar dentro de los hogares. La domótica tiene como principal objetivo la integración de objetos cotidianos con la tecnología, facilitando así su uso y control. A inicios de siglo la domótica presentaba varios inconvenientes que le impedían dar el salto al gran público, la baja capacidad de los componentes técnicos y la necesidad de realizar interconexiones cableadas entre los diferentes elementos, encareciendo mucho el coste y dificultando la instalación. En la actualidad, gracias al rápido avance de la tecnología, estos problemas han desaparecido, disponiendo en el mercado de una gran variedad de dispositivos con capacidad de computo suficientes para elementos domóticos y a bajo coste. Además, la evolución de las redes inalámbricas elimina la necesidad de hacer uso de interconexiones mediante cableado para la comunicación de información, permitiendo una sencilla instalación de dispositivos domóticos.

2.1. Situación actual.

2.1.1. Domótica en la actualidad.

Un sistema domótico está formado por un conjunto de elementos tecnológicos que permiten recoger información del entorno a partir de sensores, hacer un procesamiento de la información recibida y emitir órdenes a los actuadores o salidas. De esta manera se puede conseguir un control y automatización inteligente de las viviendas.

La domótica se ha visto muy afectada por el rápido desarrollo tecnológico, permitiendo una evolución considerable en los últimos años. Hoy en día, la domótica se puede aplicar a todo tipo de viviendas, ofreciendo diferentes soluciones dependiendo de las necesidades del cliente. Facilita el ahorro energético mediante una gestión eficiente de la iluminación, climatización, electrodomésticos o cualquier tipo de elemento eléctrico domotizado. También fomenta la accesibilidad, permitiendo y facilitando el control sobre diferentes elementos del hogar a personas con discapacidades. En general la domótica aporta confort y seguridad mediante la gestión de dispositivos y actividades domésticas. [5]

2.1.2. Investigaciones y proyectos previos sobre domótica.

La domótica está a la orden del día y por ese motivo existen multitud de investigaciones y proyectos que pretenden exprimir y desarrollar el potencial y posibilidades que ofrece la domótica. A continuación, se exponen algunos trabajos similares que pueden ayudar a enforzar el desarrollo del proyecto actual.

La entrada en vigor del Real Decreto 235/2013 el 1 de junio de 2013 que aprueba el procedimiento básico de certificación de la eficiencia energética de los edificios promueve el desarrollo de proyectos de domótica. Para mejorar la certificación energética de los edificios, se estudia la implantación de sistemas de control y automatización para los elementos de iluminación y climatización, así como el movimiento de las persianas y toldos. [6]

El protocolo X10 permite una comunicación entre dispositivos eléctricos haciendo uso de la línea eléctrica doméstica. Para conseguir una comunicación se requiere de transmisores y receptores X10, donde los pulsos de 120Khz de frecuencia durante 1ms representan el 1 en binario y la ausencia del pulso durante 1 ms indica un 0. Los pulsos se deben sincronizar en el paso por cero en la señal de la red eléctrica, donde tomará valor 0 o 1. Los dispositivos X10 no son capaces de diferenciar el paso por 0 entre una señal que va de positivo a negativo que, a la inversa, y los dos casos se interpretan de la misma manera. Además, el protocolo X10 posee redundancia en el envío de datos ya que cada pulso se transmite 2 veces, reduciendo así la posible pérdida de información. De esta manera se consigue la transmisión de datos por la línea eléctrica, y como explica Ruth Annabell en su estudio, se puede aplicar este protocolo en proyectos de control de diversos elementos domóticos sin necesidad del uso de redes inalámbricas y evitando interconexiones cableadas específicas para el sistema domótico. [7].

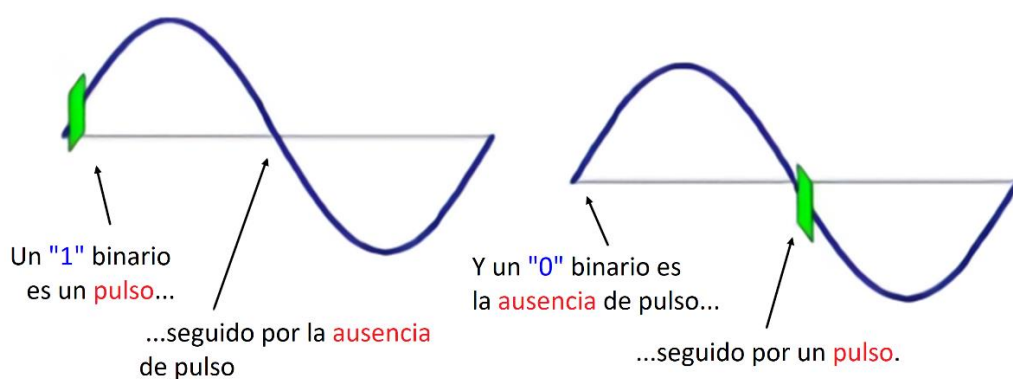


Fig. 2.1. Codificación de bits en el protocolo X10. Fuente: [8]

Angelo Geovanny ha investigado y diseñado un sistema de automatización y control de viviendas haciendo uso de la tecnología Wi-Fi. Se utiliza el microcontrolador

NodeMCU ESP8266 para la transmisión de datos y como dispositivo de control de elementos domóticos, como luces, termostatos y persianas. La comunicación y configuración de los microcontroladores se realizaría mediante una aplicación para dispositivos móviles, permitiendo su control y automatización. El proyecto utiliza componentes auxiliares conectados al microcontrolador ESP8266. Entre ellos podemos encontrar un sensor de humedad y temperatura (DTH11) con el que nutrir al sistema domótico de información ambiente que le ayude con la toma de decisiones. También posee un módulo relé con el que controlar la iluminación de 110 voltios y un módulo puente H (L298N) que permite controlar el sentido de giro de motores DC (corriente continua). [9]

2.1.3. Actualidad del mercado.

En este momento el mercado ofrece variedad de posibilidades a la hora de automatizar y controlar persianas eléctricas, cada una de ellas con sus ventajas e inconvenientes. A continuación, se recogen algunas de las opciones más destacables.

Somfy es una empresa experimentada y altamente involucrada en la investigación y desarrollo de tecnologías aplicables a casas inteligentes. Poseen gran variedad de dispositivos de control y automatización de calefacción, iluminación, puertas de garaje y, por supuesto, persianas y toldos. Además, poseen una aplicación exclusiva para cada tipo de producto que venden, así como mandos físicos para el control de estos. Los productos ofertados por Somfy son de gran calidad, aunque ello conlleva un precio excesivamente alto, sobre todo si se compara con sus alternativas de mercado. Sin embargo, su principal inconveniente es la compatibilidad de sus productos, ya que sus controles y aplicaciones móviles solo son compatibles con dispositivos “io-homecontrol” de la compañía. Por ende, si quieres automatizar una persiana eléctrica ya instalada, no sería posible, teniendo que reemplazar el kit completo de persiana eléctrica por uno de la marca. [10]

Podemos encontrar otras alternativas en el mercado como las que ofrece la empresa Maxcio. El producto que se ofrece consiste en un interruptor de persiana, modificado para poder ser utilizado mediante los comandos por voz de Alexa y Google Assistant, pudiendo optar también por una programación horaria para automatizar el sistema. Su conectividad se establece vía Wi-Fi y está destinado a redes de área local (WLAN). Como puntos fuertes se destaca la compatibilidad con la mayoría de los motores de persianas eléctricas, así como su bajo coste, que se encuentra entorno a los 30€. [11]

2.2. Diseño de soluciones.

Este trabajo pretende ofrecer una solución barata y simple para la domotización de persianas eléctricas. El producto será válido para cualquier motor o persiana eléctrica que siga las conexiones de la figura 2.2. e incluso cualquier motor de características similares con este esquema eléctrico.

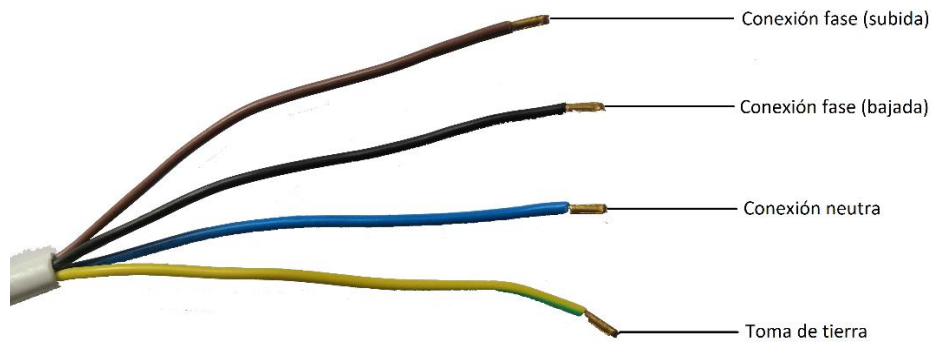


Fig. 2.2. Conexiones del motor de persiana. Fuente: Elaboración propia

La competencia del producto a desarrollar serán principalmente los dispositivos como el ofrecido por Maxcio analizado en el punto 2.1.3. Se trata de interruptores Wi-Fi con una funcionalidad similar a la que se pretende conseguir con este proyecto, permitiendo un control tanto desde el interruptor físico, como desde el dispositivo móvil, así como una automatización de los desplazamientos en diferentes días y horas. La solución que se pretende alcanzar en este proyecto tiene un gran punto diferenciador con su competencia respecto a funcionalidad, y es que el dispositivo desarrollado será capaz de conocer la altura actual de la persiana. Esto supone una gran mejora de usabilidad y funcionalidad que ofrece principalmente tres beneficios. El primero de ellos implica que se podrá visualizar desde la aplicación móvil la altura actual de cada una de las persianas domotizadas. Como segundo aspecto diferenciador, se permitirá al usuario elegir la altura de la persiana y el dispositivo al conocer la altura actual podrá moverla de forma autónoma, mientras que con los dispositivos de la competencia el usuario tendrá que estar viendo la persiana, y pararla manualmente si la quiere dejar en una altura intermedia. Por último, la competencia actual permite programar movimientos automáticos hasta el final de carrera de la persiana, esto quiere decir, hasta arriba o abajo del todo, sin embargo, la solución que propone este proyecto permitirá los desplazamientos automáticos a cualquier altura deseada por el usuario.

3. ANÁLISIS

3.1. Descripción del problema.

A continuación, se realiza un análisis general del proyecto y de los principales problemas a resolver.

3.1.1. Perspectiva del producto.

El avance constante de la tecnología ha permitido que la domótica alcance unas capacidades inimaginables hace un par de décadas, permitiendo un control prácticamente total del hogar a golpe de smartphone, incluso pudiendo manipular deferentes elementos con un simple comando de voz.

Con este proyecto se pretende ofrecer al usuario un control total de las persianas de la casa simplemente utilizando una aplicación móvil y una automatización basada en la programación horaria de los dispositivos de control de persianas. Los DCP serán fácilmente reemplazables por los interruptores comunes de las persianas eléctricas y compatibles con los sistemas de persiana motorizada genéricos, evitando así complicaciones en la instalación del producto.

Por último, se debe tener presente la minimización del coste el producto, sin dejar de lado la calidad y funcionalidad ofrecida, reduciendo el impacto económico que puede suponer la domotización de las persianas de una casa con las soluciones disponibles en el mercado actual.

3.1.2. Alcance del proyecto.

El proyecto puede ser fácilmente diferenciable en dos partes, la referida a la funcionalidad, diseño e implementación del DCP y su correspondiente en la aplicación móvil.

La aplicación permitirá a los usuarios controlar el sistema domótico de persianas de la red Wi-Fi en la que se encuentren conectados. Se podrán gestionar los DCP, permitiendo la creación de grupos para poder controlar varias persianas en una acción, evitando tener que controlar una por una. Existirá la opción de poder automatizar una persiana o grupo estableciendo una programación horaria de los DCP. La aplicación móvil estará disponible para el sistema operativo Android.

Los DCP serán capaces de controlar un motor eléctrico AC (corriente alterna) de persiana, permitiendo no solo de subir/bajar completamente la persiana, si no de establecer posiciones intermedias. El DCP debe permitir un control manual de la persiana, dando la opción al usuario de desactivar temporalmente la programación horaria. Existirán leds en el DCP que aportarán feedback al usuario sobre el estado del dispositivo.

3.2. Requisitos.

Los requisitos sirven para identificar y formalizar la funcionalidad que implementará el producto. Servirán para establecer un orden de prioridades en el desarrollo de cada una de las funcionalidades del proyecto. Además, cada requisito tendrá definidos una serie de criterios de aceptación o pruebas que permitirán validar la funcionalidad requerida una vez implementada.

3.2.1. Requisitos funcionales.

Los requisitos funcionales representan y definen los servicios que debe proporcionar el producto a desarrollar.

Se utilizará la siguiente plantilla para la definición de los requisitos funcionales.

TABLA 3.1. PLANTILLA DE REQUISITOS FUNCIONALES

Identificador: RF-DCP/APP-XY	Nombre: Nombre del requisito
Prioridad: Baja / Media / Alta	Requisitos asociados:
Descripción: Explicación de la funcionalidad que aporta el requisito. También se puede establecer lo que no debe hacer.	
Criterios de aceptación: Prueba o conjunto de ellas que permiten validar el correcto funcionamiento del requisito.	

Fuente: Elaboración propia

El identificador diferencia entre los requisitos para el dispositivo de control de persianas “DCP” y los correspondientes a la aplicación móvil “APP”. El prefijo “RF” indica que se trata de un requisito funcional. Por último, el sufijo XY serán valores decimales que representará el número del requisito. En el campo de requisitos asociados se recogen los requisitos que tienen una fuerte dependencia con el actual.

TABLA 3.2. REQUISITO FUNCIONAL RF-DCP-01

Identificador: RF-DCP-01	Nombre: Control manual
Prioridad: Alta	Requisitos asociados: RF-DCP-03
Descripción: El DCP permite el control mediante botones físicos para subir o bajar la persiana. La persiana se moverá mientras dure la pulsación del botón.	
Criterios de aceptación: 1) Presionar el botón para subir la persiana. La persiana se debe subir mientras se mantenga el botón pulsado o se alcance el límite superior. 2) Presionar el botón para bajar la persiana. La persiana se debe bajar mientras se mantenga el botón pulsado o se alcance el límite inferior.	

Fuente: Elaboración propia

TABLA 3.3. REQUISITO FUNCIONAL RF-DCP-02

Identificador: RF-DCP-02	Nombre: Conexión a red Wi-Fi
Prioridad: Alta	Requisitos asociados: RF-DCP-04, RF-DCP-10, RF-APP-01
Descripción: El DCP recibe los datos de acceso a una red Wi-Fi. Se intentará establecer la conexión y se informará al emisor de los datos de acceso con el resultado.	
Criterios de aceptación: 1) Se reciben datos de una red Wi-Fi. La conexión se establece correctamente y se informa de ello al emisor. 2) Se reciben datos de una red Wi-Fi. No es posible establecer la conexión, se informa de ello al emisor.	

Fuente: Elaboración propia

TABLA 3.4. REQUISITO FUNCIONAL RF-DCP-03

Identificador: RF-DCP-03	Nombre: Sincronización de altura
Prioridad: Alta	Requisitos asociados: RF-DCP-01, RF-APP-12
<p>Descripción: Cuando se inicia el proceso de sincronización en altura el DCP subirá la persiana, parando al recibir el aviso de límite superior alcanzado. Tras el primer aviso la persiana comenzará a bajar y parará al recibir el aviso de límite inferior alcanzado. Este proceso permite al DCP conocer la altura a la que se encuentra la persiana (incluido tras realizar movimientos) y el tiempo que requiere un movimiento de extremo a extremo.</p>	
<p>Criterios de aceptación: 1) Realizar un proceso de sincronización. Comprobar que el tiempo calculado por el DCP coincide con el intervalo de tiempo entre los avisos de límite superior e inferior alcanzados.</p> <p>2) Realizar un proceso de sincronización. Comprobar que la posición de la persiana almacenada en el DCP es 0% de altura (totalmente bajada).</p>	

Fuente: Elaboración propia

TABLA 3.5. REQUISITO FUNCIONAL RF-DCP-04

Identificador: RF-DCP-04	Nombre: Atender orden vía Wi-Fi
Prioridad: Alta	Requisitos asociados: RF-DCP-02, RF-DCP-05, RF-DCP-06, RF-DCP-09, RF-DCP-12, RF-APP-07, RF-DCP-07
<p>Descripción: El DCP atenderá las solicitudes de conexión vía Wi-Fi para la transmisión de datos. El DCP recibirá los datos enviados por el cliente, que deben componerse de una orden (tipo de acción solicitada) y datos adicionales que pueden ser necesarios para la realización de la acción.</p> <p>Si los datos recibidos cumplen las características explicadas se ejecutará la orden especificada y se informará al cliente que la petición se ha recibido correctamente.</p>	
<p>Criterios de aceptación: 1) Se reciben datos de un cliente vía Wi-Fi para la realización de una acción. Los datos deben ser correctos. Se debe ejecutar correctamente la orden solicitada e informar de ello al cliente.</p> <p>2) Se reciben datos de un cliente vía Wi-Fi que no cumplen la estructura definida. No se realiza ninguna acción.</p> <p>3) Se reciben datos de un cliente vía Wi-Fi para la realización de una acción. La estructura de los datos es correcta, pero la orden solicitada no existe. No se realiza ninguna acción.</p>	

Fuente: Elaboración propia

TABLA 3.6. REQUISITO FUNCIONAL RF-DCP-05

Identificador: RF-DCP-05	Nombre: Atender orden de altura
Prioridad: Alta	Requisitos asociados: RF-DCP-04, RF-APP-05
Descripción: Se recibe una orden para desplazar la persiana a una nueva altura. El DCP determinará si la persiana debe subir o bajar y durante cuánto tiempo. Tras realizar el movimiento de la persiana el DCP avisa al emisor con la nueva altura alcanzada.	
Criterios de aceptación: 1) Realizar 5 pruebas donde la persiana debe alcanzar una posición superior. Verificar el correcto desplazamiento de la persiana. 2) Realizar 5 pruebas donde la persiana debe alcanzar una posición inferior. Verificar el correcto desplazamiento de la persiana. 3) Realizar una prueba donde la altura de destino se la actual. La persiana no debe moverse.	

Fuente: Elaboración propia

TABLA 3.7. REQUISITO FUNCIONAL RF-DCP-06

Identificador: RF-DCP-06	Nombre: Programación horaria
Prioridad: Alta	Requisitos asociados: RF-DCP-07, RF-DCP-08, RF-APP-08, RF-DCP-04
Descripción: El DCP almacena un horario con desplazamientos que debe realizar la persiana. Al alcanzar la hora de un desplazamiento se efectuará el movimiento correspondiente de la persiana para alcanzar la posición determinada.	
Criterios de aceptación: 1) Establecer una programación horaria (explicada en RF-DCP-07). Verificar que la persiana se desplaza a la altura establecida cuando se alcanza una hora determinada en la programación horaria del DCP.	

Fuente: Elaboración propia

TABLA 3.8. REQUISITO FUNCIONAL RF-DCP-07

Identificador: RF-DCP-07	Nombre: Modificar programación horaria
Prioridad: Alta	Requisitos asociados: RF-DCP-04, RF-DCP-06, RF-APP-06
Descripción: Se recibe una orden para modificar la programación horaria. El DCP modificará su programación horaria en función de los nuevos datos.	
Criterios de aceptación: 1) Se recibe una orden para modificar la programación horaria. Comprobar que la nueva configuración de la programación horaria se almacena correctamente en la memoria del dispositivo. 2) Se recibe una orden para modificar la programación horaria que no cumple con la estructura correcta. No se modifica la programación horaria.	

Fuente: Elaboración propia

TABLA 3.9. REQUISITO FUNCIONAL RF-DCP-08

Identificador: RF-DCP-08	Nombre: Cambiar modo funcionamiento (botón físico)
Prioridad: Baja	Requisitos asociados: RF-DCP-06, RF-DCP-10
Descripción: La activación de un botón físico en el DCP permite cambiar el modo de funcionamiento del dispositivo, alternando entre “solo manual” y “manual + programación horaria”. Esta opción permite al usuario activar y desactivar la programación horaria.	
Criterios de aceptación: 1) El modo inicial debe ser “manual + automático”. Activar el botón físico para pasar a modo “solo manual”. Verificar que la programación horaria establecida no modifica la altura de la persiana. 2) El modo inicial debe ser “solo manual”. Activar el botón físico para pasar a modo “manual + automático”. Verificar que la programación horaria establecida sí modifica la altura de la persiana.	

Fuente: Elaboración propia

TABLA 3.10. REQUISITO FUNCIONAL RF-DCP-09

Identificador: RF-DCP-09	Nombre: Cambiar modo funcionamiento (orden Wi-Fi)
Prioridad: Baja	Requisitos asociados: RF-DCP-04, RF-APP-10, RF-DCP-10
Descripción: Se recibe una orden para establecer el modo de funcionamiento del DCP (“solo manual” o “manual + programación horaria”). Esta opción permite al usuario activar y desactivar la programación horaria.	
Criterios de aceptación: 1) El modo inicial debe ser “manual + automático”. Se recibe una orden para cambiar de modo a “solo manual”. Verificar que la programación horaria establecida no modifica la altura de la persiana. 2) El modo inicial debe ser “solo manual”. Se recibe una orden para cambiar de modo a “manual + automático”. Verificar que la programación horaria establecida sí modifica la altura de la persiana.	

Fuente: Elaboración propia

TABLA 3.11. REQUISITO FUNCIONAL RF-DCP-10

Identificador: RF-DCP-10	Nombre: LED de estado
Prioridad: Baja	Requisitos asociados: RF-DCP-02, RF-DCP-08, RF-DCP-09
Descripción: El DCP contará con diodos led que aportarán información al usuario respecto al modo de funcionamiento establecido (manual / manual + automático) y sobre el estado de la conexión Wi-Fi.	
Criterios de aceptación: 1) Realizar una prueba con todas las combinaciones de estados del DCP y verificar que los leds transmiten un feedback correcto, según el comportamiento definido.	

Fuente: Elaboración propia

TABLA 3.12. REQUISITO FUNCIONAL RF-DCP-11

Identificador: RF-DCP-11	Nombre: Reset
Prioridad: Media	Requisitos asociados:
Descripción: Un botón físico en el DCP permite borrar toda la configuración almacenada en el dispositivo. Tras pulsar el botón de reset el DCP regresará a su estado original, por lo que será necesario reconfigurar la red Wi-Fi o volver a sincronizar la altura de la persiana.	
Criterios de aceptación: 1) Se presiona el botón de reset. Comprobar que el DCP ha eliminado la configuración Wi-Fi. 2) Se presiona el botón de reset. Comprobar que el DCP ha eliminado la configuración de programación horaria. 3) Se presiona el botón de reset. Comprobar que el DCP ha perdido la sincronización de altura de la persiana.	

Fuente: Elaboración propia

TABLA 3.13. REQUISITO FUNCIONAL RF-DCP-12

Identificador: RF-DCP-12	Nombre: Actualizar fecha y hora
Prioridad: Alta	Requisitos asociados: RF-DCP-04, RF-APP-11
Descripción: El DCP almacena una fecha y hora que podrá ser actualizada por el usuario.	
Criterios de aceptación: 1) Recibir una solicitud de cambio de fecha y hora. Se debe verificar que la nueva fecha y hora del DCP concuerda con la enviada.	

Fuente: Elaboración propia

TABLA 3.14. REQUISITO FUNCIONAL RF-APP-01

Identificador: RF-APP-01	Nombre: Añadir DCP
Prioridad: Alta	Requisitos asociados: RF-DCP-02, RF-APP-05
Descripción: Un nuevo DCP no tendrá los datos de acceso para conectarse a la red Wi-Fi. La aplicación debe permitir añadir un nuevo DCP a la red Wi-Fi. Se transmitirán al nuevo DCP los datos de acceso a la red Wi-Fi (SSID y contraseña).	
Criterios de aceptación: 1) Seleccionar la opción “Añadir DCP” de la aplicación. Comprobar que el dispositivo móvil ha enviado correctamente los datos de acceso de la red Wi-Fi.	

Fuente: Elaboración propia

TABLA 3.15. REQUISITO FUNCIONAL RF-APP-02

Identificador: RF-APP-02	Nombre: Mostrar DCP
Prioridad: Alta	Requisitos asociados: RF-APP-08
Descripción: La aplicación mostrará un listado con los DCP conectados en la red Wi-Fi.	
Criterios de aceptación: 1) La prueba se debe realizar con más de 1 DCP conectado a la red Wi-Fi. La aplicación debe mostrar en el listado todos los DCP conectados en la red Wi-Fi.	

Fuente: Elaboración propia

TABLA 3.16. REQUISITO FUNCIONAL RF-APP-03

Identificador: RF-APP-03	Nombre: Crear un grupo
Prioridad: Media	Requisitos asociados: RF-APP-04, RF-APP-05
Descripción: Se permitirá crear un grupo de dispositivos DCP ya existentes. Un grupo permite controlar y establecer una programación horaria sobre el conjunto de DCP que lo forman. Un grupo se controlará como si de un DCP individual se tratase.	
Criterios de aceptación: 1) Crear un grupo de 2 o más DCP. Controlar la altura de las persianas del grupo (RF-APP-05) y verificar que todas las persianas realizan el desplazamiento. 2) Crear un grupo de 2 o más DCP. Modificar la programación horaria de las persianas del grupo (RF-APP-06) y verificar que todas las persianas realizan el desplazamiento de la nueva programación horaria.	

Fuente: Elaboración propia

TABLA 3.17. REQUISITO FUNCIONAL RF-APP-04

Identificador: RF-APP-04	Nombre: Eliminar un grupo
Prioridad: Baja	Requisitos asociados: RF-APP-03
Descripción: Se permitirá borrar un grupo de DCP creando anteriormente. Se borrará la programación horaria establecida en cada DCP correspondiente al grupo.	
Criterios de aceptación: 1) Eliminar un grupo formado por 2 o más DCP. El grupo debe desaparecer de la aplicación. Verificar que la configuración del grupo se borra en los DCP.	

Fuente: Elaboración propia

TABLA 3.18. REQUISITO FUNCIONAL RF-APP-05

Identificador: RF-APP-05	Nombre: Controlar un DCP
Prioridad: Alta	Requisitos asociados: RF-APP-01, RF-APP-03, RF-DCP-05
Descripción: La aplicación permite seleccionar un DCP de la lista. Se permite controlar la altura de la persiana asociada al DCP. Un grupo de DCP se controlará como si de un DCP individual se tratase.	
Criterios de aceptación: 1) Seleccionar el DCP asociado a la persiana que se desea controlar. Modificar la altura de la persiana. Comprobar que la persiana se desplaza a la altura establecida.	

Fuente: Elaboración propia

TABLA 3.19. REQUISITO FUNCIONAL RF-APP-06

Identificador: RF-APP-06	Nombre: Programación horaria DCP
Prioridad: Alta	Requisitos asociados: RF-DCP-07
Descripción: La aplicación permite seleccionar un DCP de la lista. Se permite modificar la programación horaria de la persiana asociada al DCP. A un grupo de DCP se le modificará la programación horaria como si de un DCP individual se tratase.	
Criterios de aceptación: 1) Seleccionar el DCP asociado a la persiana sobre el que se desea modificar la programación horaria. Modificar la programación horaria del DCP. Comprobar que la persiana se desplaza según la programación horaria establecida.	

Fuente: Elaboración propia

TABLA 3.20. REQUISITO FUNCIONAL RF-APP-07

Identificador: RF-APP-07	Nombre: Editar nombre
Prioridad: Baja	Requisitos asociados: RF-DCP-04
Descripción: Cada DCP o grupo tiene un nombre asociado. Permite modificar el nombre de un DCP o grupo por uno nuevo.	
Criterios de aceptación: 1) Seleccionar un DCP y elegir la opción de editar nombre. Introducir un nuevo nombre y verificar que se actualiza correctamente. 2) Seleccionar un grupo y elegir la opción de editar nombre. Introducir un nuevo nombre y verificar que se actualiza correctamente.	

Fuente: Elaboración propia

TABLA 3.21. REQUISITO FUNCIONAL RF-APP-08

Identificador: RF-APP-08	Nombre: Visualizar estado
Prioridad: Alta	Requisitos asociados: RF-APP-02, RF-DCP-06
Descripción: Al seleccionar un DCP o grupo se visualizará información relevante sobre el estado actual y configuración. Esta información incluye nombre del dispositivo/grupo, posición actual de la persiana y la configuración de la programación horaria.	
Criterios de aceptación: 1) La prueba se debe realizar con más de 1 DCP conectado a la red Wi-Fi. La aplicación debe mostrar en el listado todos los DCP conectados en la red Wi-Fi.	

Fuente: Elaboración propia

TABLA 3.22. REQUISITO FUNCIONAL RF-APP-10

Identificador: RF-APP-10	Nombre: Cambiar modo
Prioridad: Baja	Requisitos asociados: RF-DCP-09
Descripción: Permite modificar el modo de funcionamiento de un DCP, pudiendo alternar entre programación horaria activada o desactivada.	
Criterios de aceptación: 1) Seleccionar un DCP y desactivar la programación horaria. Verificar que no se aplican los desplazamientos de la configuración horaria establecida. 2) Seleccionar un DCP y activar la programación horaria. Verificar que se aplican los desplazamientos de la configuración horaria establecida.	

Fuente: Elaboración propia

TABLA 3.23. REQUISITO FUNCIONAL RF-APP-11

Identificador: RF-APP-11	Nombre: Actualizar fecha y hora
Prioridad: Alta	Requisitos asociados: RF-DCP-12
Descripción: Permite actualizar la fecha y hora de un DCP seleccionado a la fecha y hora del dispositivo móvil.	
Criterios de aceptación: 1) Seleccionar un DCP y elegir la opción de actualizar fecha y hora. Verificar que se envía una petición para actualizar la fecha y hora con los datos correctos.	

Fuente: Elaboración propia

TABLA 3.24. REQUISITO FUNCIONAL RF-APP-12

Identificador: RF-APP-12	Nombre: Sincronizar desplazamiento
Prioridad: Alta	Requisitos asociados: RF-DCP-03
Descripción: La altura de la persiana se mide según su tiempo de subida y bajada. Se podrá ajustar este tiempo para cada dispositivo y persiana mediante la aplicación móvil.	
Criterios de aceptación: 1) Realizar la sincronización de desplazamiento con la persiana. Verificar que los tiempos medidos por la aplicación concuerdan con los reales.	

Fuente: Elaboración propia

3.2.2. Matriz de trazabilidad.

TABLA 3.25. MATRIZ DE TRAZABILIDAD

		RF-DCP												RF-APP											
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	10	11	12	
RF-DCP	1																								
	2																								
	3																								
	4																								
	5																								
	6																								
	7																								
	8																								
	9																								
	10																								
	11																								
	12																								
RF-APP	1																								
	2																								
	3																								
	4																								
	5																								
	6																								
	7																								
	8																								
	10																								
	11																								
	12																								

Fuente: Elaboración propia

3.2.3. Requisitos no funcionales.

Las propiedades y restricciones del producto vienen definidas por los requisitos no funcionales. En este caso el prefijo “RNF” lo identifica como un requisito no funcional, seguido del número del requisito.

TABLA 3.26. REQUISITOS NO FUNCIONALES

Identificador	Nombre	Descripción
RNF-01	Persistencia de datos	Los datos referentes a la configuración del DCP deben ser persistentes almacenándose en la memoria flash del dispositivo. Evitará pérdida de información frente a cortes de energía.
RNF-02	Reconexión Wi-Fi	Si el DCP pierde la conexión a la red Wi-Fi configurada, se intentará reconectar cada intervalo de 5 segundos.
RNF-03	Posición de la persiana	Se utilizará el tiempo de subida/bajada de la persiana para conocer su posición exacta en cada momento.
RNF-04	Número de grupos	Cada dispositivo podrá pertenecer a un máximo de 10 grupos.
RNF-05	Número de desplazamientos	Por cada grupo se podrán programar un máximo de 20 desplazamientos automáticos.
RNF -06	Peticiones Wi-Fi	El envío de peticiones Wi-Fi entre el DCP y la aplicación se realizará mediante el protocolo TCP.

Fuente: Elaboración propia

3.3. Casos de uso.

En esta sección se identifican los casos de uso, los cuales describirán las interacciones posibles del usuario con el sistema.

3.3.1. Diagrama de casos de uso.

En la figura 3.1. se muestra el diagrama que recoge los casos de uso contemplados en el sistema. En este caso, solo existe un único actor, identificado como el usuario.

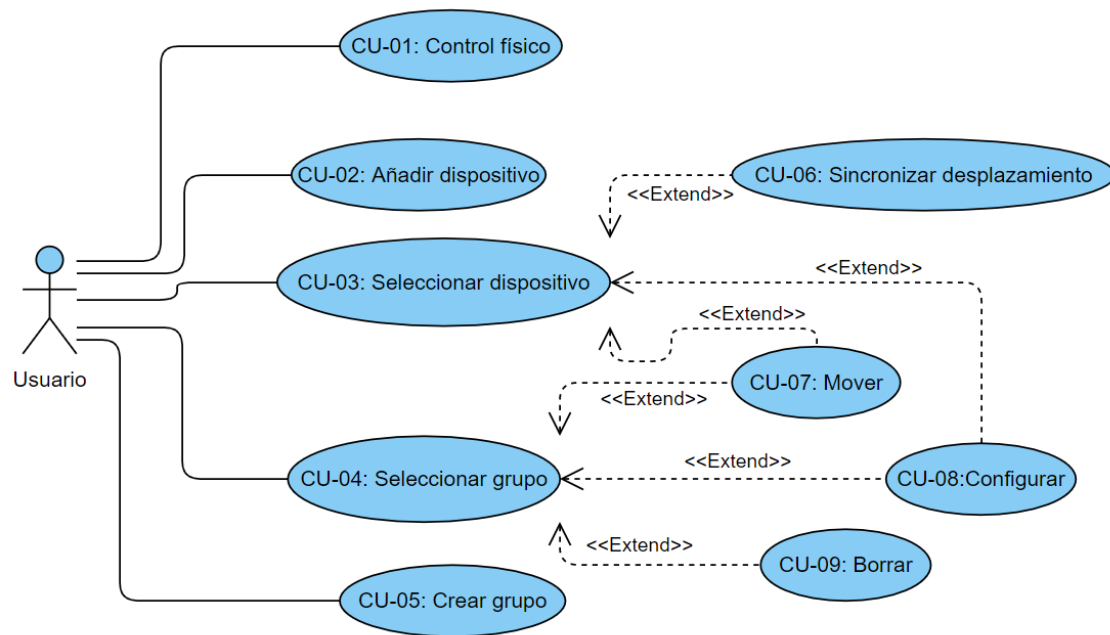


Fig. 3.1. Diagrama de casos de uso. Fuente: Elaboración propia

3.3.2. Casos de uso de alto nivel.

En la especificación de los casos de uso de alto nivel, además del identificador, el nombre y los actores que interactúan con el caso de uso, también se incluye una breve descripción, así como las precondiciones y postcondiciones que se deben dar.

TABLA 3.27. CASO DE USO CU-01

Identificador	CU-01
Nombre	Control físico
Actores	Usuario
Descripción	Permite controlar la altura de la persiana mediante botones físicos.
Precondiciones	1. El dispositivo de control debe estar instalado en una persiana.
Postcondiciones	1. La persiana se desplaza hacia arriba mientras se pulsa el botón de subir. 2. La persiana se desplaza hacia abajo mientras se pulsa el botón de bajar.

Fuente: Elaboración propia

TABLA 3.28. CASO DE USO CU-02

Identificador	CU-02
Nombre	Añadir dispositivo
Actores	Usuario
Descripción	Añade a la red Wi-Fi un dispositivo de control.
Precondiciones	1. El dispositivo de control no puede tener guardado los datos de otra red Wi-Fi.
Postcondiciones	1. El dispositivo estará conectado a la red Wi-Fi. El led de estado Wi-Fi debe estar encendido.

Fuente: Elaboración propia

TABLA 3.29. CASO DE USO CU-03

Identificador	CU-03
Nombre	Seleccionar dispositivo
Actores	Usuario
Descripción	Permite seleccionar un dispositivo perteneciente a la red Wi-Fi mostrando su configuración y estado actual.
Precondiciones	1. El dispositivo de control debe estar conectado a la red Wi-Fi.
Postcondiciones	1. Se muestran la información actual del dispositivo y su programación horaria.

Fuente: Elaboración propia

TABLA 3.30. CASO DE USO CU-04

Identificador	CU-04
Nombre	Seleccionar grupo
Actores	Usuario
Descripción	Permite seleccionar un grupo existente, mostrando los dispositivos asignados y la configuración.
Precondiciones	1. El grupo debe ser creado previamente.
Postcondiciones	1. Se muestran la información actual del grupo, los dispositivos asociados y su programación horaria.

Fuente: Elaboración propia

TABLA 3.31. CASO DE USO CU-05

Identificador	CU-05
Nombre	Crear grupo
Actores	Usuario
Descripción	Crea un nuevo grupo para controlar un conjunto de dispositivos con el nombre introducido por el usuario.
Precondiciones	1. No puede existir un grupo con ese nombre.
Postcondiciones	1. Aparecerá un nuevo grupo vacío, con el nombre introducido por el usuario.

Fuente: Elaboración propia

TABLA 3.32. CASO DE USO CU-06

Identificador	CU-06
Nombre	Sincronizar desplazamiento
Actores	Usuario
Descripción	Calcula y actualiza los tiempos de desplazamiento de la persiana.
Precondiciones	
Postcondiciones	1. Se deben actualizar los tiempos de desplazamiento de la persiana.

Fuente: Elaboración propia

TABLA 3.33. CASO DE USO CU-07

Identificador	CU-07
Nombre	Mover
Actores	Usuario
Descripción	Permite mover a una nueva altura un dispositivo o un grupo de ellos a la altura seleccionada.
Precondiciones	1. Seleccionar un dispositivo o grupo.
Postcondiciones	1. Se deben desplazar a la nueva altura las persianas afectadas.

Fuente: Elaboración propia

TABLA 3.34. CASO DE USO CU-08

Identificador	CU-08
Nombre	Configurar
Actores	Usuario
Descripción	Permite la modificación del nombre del dispositivo o grupo y su programación horaria.
Precondiciones	1. Seleccionar un dispositivo o grupo.
Postcondiciones	1. Se guardará la configuración establecida.

Fuente: Elaboración propia

TABLA 3.35. CASO DE USO CU-09

Identificador	CU-09
Nombre	Borrar
Actores	Usuario
Descripción	Borra el grupo seleccionado y su configuración en los dispositivos.
Precondiciones	1. Seleccionar un grupo.
Postcondiciones	1. El grupo desaparecerá. 2. Se borrará la configuración del grupo de los dispositivos.

Fuente: Elaboración propia

4. DISEÑO

En la sección de diseño se estudia y realiza la propuesta de trabajo. Se lleva a cabo una tarea de investigación de las diferentes alternativas de diseño y se argumenta cada decisión tomada en función de los beneficios para alcanzar una buena solución final.

4.1. Descripción de la arquitectura.

El sistema es fácilmente divisible en dos evolutivos, por lo que se ha optado por una arquitectura donde el componente DCP ofrecerá una interfaz a la aplicación, la cual, mediante peticiones podrá gestionar y controlar las persianas de la red Wi-Fi.

4.2. Diagrama de clases.

El diagrama de clases describe la estructura del sistema mediante las clases y sus relaciones. Cada clase puede tener atributos y métodos que podrán ser de carácter público o privado.

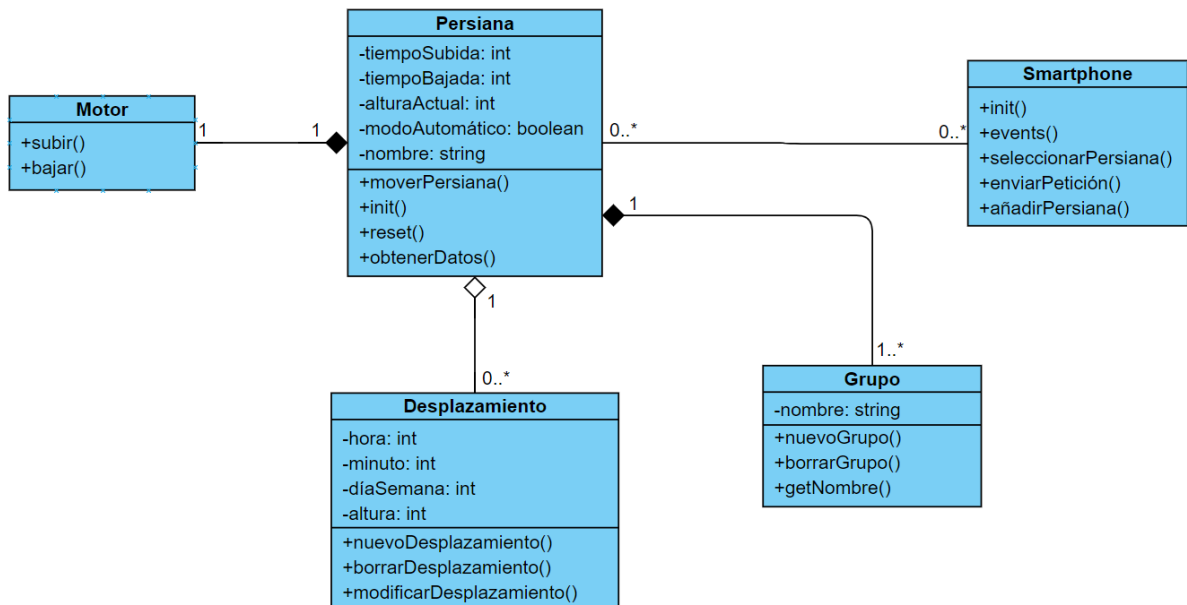


Fig. 4.1. Diagrama de clases. Fuente: Elaboración propia

4.3. Diagrama de componentes.

En este tipo de diagramas cada uno de los componentes se define como una pieza ejecutable de software que compondrá el sistema. El componente DCP estará acoplado a la persiana y ofrecerá una interfaz que permita a otros componentes acceder a la información y modificar la posición y configuración de la persiana. El gestor de persianas y el gestor de grupos ofrecen una interfaz al controlador principal, facilitando la organización y manejo de las persianas que componen la red. El componente vista servirá para ofrecer al usuario una interfaz con la que interactuar con el sistema. EspTouch se trata del componente software que permitirá la transmisión de las credenciales Wi-Fi con un nuevo dispositivo.

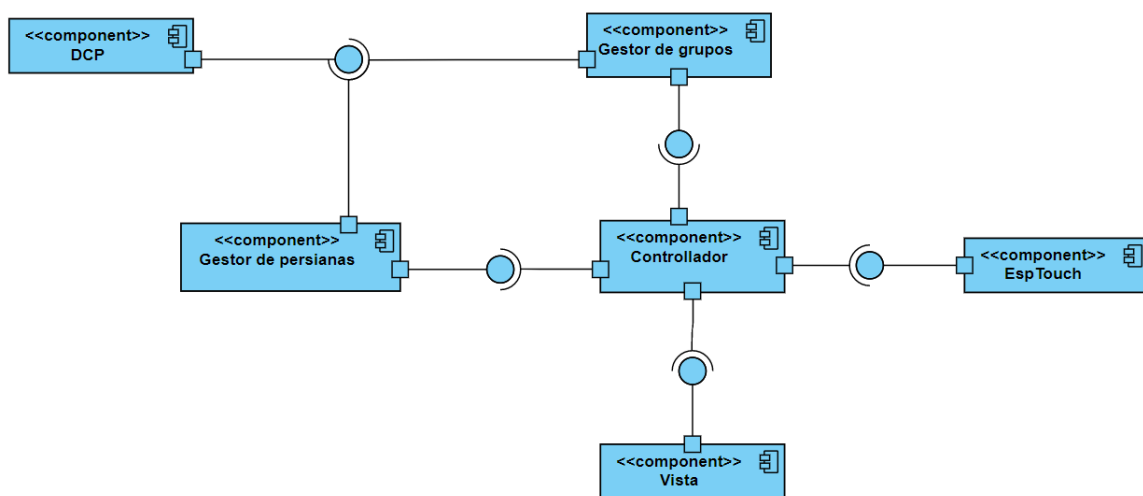


Fig. 4.2. Diagrama de componentes. Fuente: Elaboración propia.

4.4. Descripción del esquema de comunicaciones.

El esquema de comunicaciones está constituido por tres diferentes situaciones de transmisión de datos que se pueden dar entre el dispositivo móvil y el DCP.

En primer lugar, una transmisión de datos con el DCP como emisor y los dispositivos móviles que ejecuten la aplicación como destinatario objetivo. Se utilizará el protocolo UDP para la emisión de paquetes de difusión a través de la red mediante la técnica de broadcast. El uso del broadcast es necesario para que los dispositivos móviles puedan identificar a los DCP por primera vez, ya que al principio no conocerán sus direcciones IP.

El segundo tipo de comunicaciones se basará en la realización de peticiones por parte de la aplicación móvil a los DCP, que actuarán como servidores. Este caso será el que desempeñe el principal flujo de información entre los dos módulos de la arquitectura y el que permitirá el control de la persiana desde el dispositivo móvil. Para ello se

utilizará un protocolo TCP, que ofrece una comunicación fiable de manera simple, estableciendo una conexión entre el dispositivo móvil y el DCP hasta que finalice el flujo de datos requerido por la petición de la aplicación.

Por último, cuando se quiere añadir un nuevo DCP se le tendrán que enviar las credenciales de la red Wi-Fi, ya que sin ellos no se podrá establecer el resto de las comunicaciones. Para llevar a cabo esta transmisión de datos, el DCP abrirá un punto de acceso Wi-Fi al cual se conectará el dispositivo móvil y le transmitirá las credenciales de forma segura, cifrando los datos.

4.5. Diseño del DCP.

El dispositivo de control de persianas o DCP es el componente físico que permitirá controlar y automatizar la persiana motorizada. El diseño del componente requiere tanto de un estudio de los componentes hardware, como de la implementación software, para conseguir los requisitos de funcionamiento establecidos.

4.5.1. Microcontrolador.

El componente principal del DCP será un microcontrolador, el cual permita manejar los componentes auxiliares del proyecto. Se debe tener presente que la conectividad Wi-Fi del microcontrolador es una necesidad vital para que el sistema funcione y actualmente el mercado ofrece muchas alternativas de microcontroladores con antena Wi-Fi integrada, por lo que no es necesario un módulo extra para lograr la conexión. Existen muchos microcontroladores enfocados a proyectos IoT que cumplen los requerimientos del proyecto, y a continuación veremos el seleccionado para el desarrollo del proyecto, así como otras alternativas.

ESP32:

El dispositivo ESP32 se trata de un microcontrolador diseñado por la compañía Espressif Systems y está muy extendido en los desarrollos de proyectos IoT. La principal ventaja de este microcontrolador es el fácil control de la conectividad Wi-Fi, permitiendo el control de componentes auxiliares mediante sus pines de entrada y salida. Como características principales podemos destacar su conectividad Wi-Fi y Bluetooth, un procesador de doble núcleo y criptografía acelerada por hardware. En la siguiente tabla se detallan sus especificaciones.

TABLA 4.1. ESPECIFICACIONES ESP32

Característica	Descripción
Procesador	Xtensa LX6 de 32 bits (doble núcleo)
Velocidad de reloj	160 MHz (240 MHz con overclock)
Voltaje de funcionamiento	3.3V
Memoria flash	16MB
Memoria SRAM	520KB
Wi-Fi	802.11 b/g/n
Bluetooth	v4.2 BR/EDR + BLE
Pines de E/S (GPIO)	36
Corriente suministrada (valor máximo)	1200mA
Consumo de corriente (valor máximo)	180mA

Fuente: [12]

Como se ha comentado antes, también posee hardware criptográfico dedicado a la aceleración de AES, SHA-2, ECC y RNG. Además, realiza un cifrado de la memoria flash del dispositivo.

Si bien el voltaje de funcionamiento del dispositivo es de 3.3V, cuenta con una salida de 5V capaz de alimentar componentes externos.

NodeMCU ESP8266:

El ESP8266 se trata de un chip Wi-Fi de bajo costo, también fabricado por Espressif Systems en el año 2013. Sirve como módulo auxiliar para los microcontroladores, permitiendo así su conexión a un red Wi-Fi y la transmisión de datos. El potencial de este módulo llevó al desarrollo de la versión NodeMCU ESP8266 en 2014, que integraba el chip Wi-Fi en el propio microcontrolador y consiguiendo así una comunicación inalámbrica más sencilla. Este dispositivo es el predecesor del ya comentado ESP32.

TABLA 4.2. ESPECIFICACIONES NODEMCU ESP8266

Característica	Descripción
Procesador	Xtensa LX106 de 32 bits
Velocidad de reloj	80 MHz (160 MHz con overclock)
Voltaje de funcionamiento	3.3V
Memoria flash	4MB
Memoria SRAM	64KB
Wi-Fi	802.11 b/g/n
Pines de E/S (GPIO)	16
Corriente suministrada (valor máximo)	1000mA
Consumo de corriente (valor máximo)	120mA

Fuente: [13]

Los dos microcontroladores analizados cuentan con capacidad suficiente para alcanzar la funcionalidad propuesta en el proyecto. Ambos dispositivos se pueden adquirir por un precio inferior a los 10€, pudiéndose encontrar mucho más barato en función del ensamblador y el mercado donde se adquiera. Si bien el proyecto intenta alcanzar los objetivos propuestos creando un producto al menor coste posible, en este caso se optará por el dispositivo ESP32, que siendo ligeramente más caro (entorno a un 20%), cuenta con unas especificaciones notablemente superiores y funcionalidades adicionales. La elección del dispositivo con las mejores características deja un gran margen para permitir actualizaciones futuras y mejora de la funcionalidad sin necesidad de modificar el hardware.

De este modo queda el ESP32 como el candidato seleccionado para desarrollo del proyecto, dejando al NodeMCU ESP8266 como una alternativa de diseño totalmente viable.

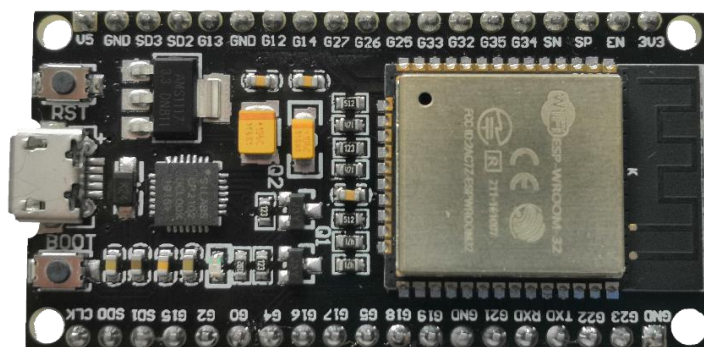


Fig. 4.3. Microcontrolador ESP32. Fuente: Elaboración propia

4.5.2. Módulo relé.

El microcontrolador es capaz de ofrecer hasta 5V para alimentar pequeños componentes externos en corriente continua. Los motores de persiana comunes suelen ser alimentados por la toma de corriente del hogar, en el caso de España, 230V de corriente alterna a 50Hz. El microcontrolador debe ser capaz de controlar el motor eléctrico, por lo que la mejor opción en este caso es utilizar un relé, que, controlado desde el microcontrolador, se puede manejar el paso de corriente de un circuito eléctrico a 230V AC.

Relé electromagnético:

El funcionamiento de un relé electromagnético se basa en una bobina y un electroimán. El relé posee una conexión de entrada que está directamente conectada a una conexión de salida, a la cual se le llama “contacto normalmente cerrado”. Cuando se permite el paso de corriente por la bobina, ésta se magnetiza y desplaza un componente móvil abriendo el circuito entre la conexión de entrada y la normalmente cerrada, y cerrando el circuito con el contacto normalmente abierto.

Relé de estado sólido:

Al igual que los relés electromagnéticos, los relés de estado sólido actúan como interruptores que controlan el paso de electricidad. En este caso, se utilizan semiconductores de potencia que permiten conmutar la corriente eléctrica. Comparado con los relés electromagnéticos, los relés de estado sólido no cuentan con piezas móviles, lo que les permite conmutar a velocidades del orden de milisegundos y un funcionamiento silencioso. Además, no existen contactos mecánicos que se puedan desgastar, los que les da una mayor vida útil.

Si bien los relés de estado sólido poseen unas características superiores a los relés electromagnéticos, tienen un precio considerablemente superior. También es importante tener en cuenta que las principales ventajas de los relés de estado sólido no suponen prácticamente ningún beneficio en el contexto de este proyecto. El motor eléctrico de la persiana ya emite sonido, por lo que el pequeño chasquido que produce el relé electromagnético al activarse será indiferente. Además, al tratarse de controlar una persiana eléctrica el número de conmutaciones que realizará el relé son lo suficientemente bajas como para que el desgaste y la vida útil no supongan un problema. Todos estos motivos convierten al relé electromagnético en una mejor opción para el desarrollo del proyecto.

Conexiones del motor de persiana:

Un motor de persiana común posee 4 cables para las conexiones. Un primer cable para la toma de tierra, otro cable que se conectará a la fase de la instalación eléctrica y, por último, dos cables que irán conectados a la conexión neutra cuando se quiera realizar el movimiento de la persiana, uno para cada sentido de giro del motor. Estas conexiones se mostraron en la figura 2.2.

Debido a la necesidad de poder controlar el paso de corriente por 2 cables, para el movimiento de subida de la persiana y el de bajada, se ha optado por utilizar un módulo relé de 2 canales.

TABLA 4.3. ESPECIFICACIONES MÓDULO RELÉ

Característica	Descripción
Voltaje de funcionamiento	5V
Número de canales	2
Capacidad máxima soportada	250V(AC) – 10A / 30V(DC) – 10A
Modo de activación	Activo por nivel bajo
Tiempo de acción	10ms
Consumo de corriente (por bobina)	90mA
Consumo de corriente (por señal de control)	10mA

Fuente: Elaboración propia



Fig. 4.4. Módulo relé de dos canales. Fuente: Elaboración propia

4.5.3. Módulo RTC.

Un reloj en tiempo real o RTC es un componente electrónico capaz de mantener la hora actual. Gracias a su bajo consumo y la existencia de una alimentación externa (como una pila de botón) es tolerante a cortes de corriente.

El DCP permite establecer una programación horaria con el que automatizar el movimiento de la persiana en función de la hora, por lo que es necesario que el microcontrolador conozca la hora actual. El microcontrolador es capaz de mantener la hora actual mientras esté encendido, pero si se produce un corte en el suministro eléctrico, el microcontrolador perderá el valor horario. Haciendo uso de un módulo RTC el microcontrolador podrá recuperar el valor de la fecha y hora actual, pudiendo continuar con su funcionamiento normal tras recuperar el suministro eléctrico.

El módulo RTC seleccionado es el DS3231, tratándose de un reloj de alta precisión que cuenta con un oscilador de cristal con compensación de temperatura. La comunicación con el microcontrolador será a través de la interfaz I²C.

TABLA 4.4. ESPECIFICACIONES MÓDULO DS3231

Característica	Descripción
Voltaje de funcionamiento	De 3V a 5V
Error	±2ppm operando entre 0°C y 40°C
Batería de respaldo	Sí
Consumo de corriente en reposo	170µA
Consumo de corriente con el bus I ² C activado	300µA

Fuente: [14]



Fig. 4.5. Módulo RTC DS3231. Fuente: Elaboración propia

4.5.4. Consumo de corriente del DCP.

Un microcontrolador puede suministrar la corriente eléctrica a los componentes auxiliares que tenga conectados, pero se debe tener en cuenta que cada microcontrolador tiene una capacidad máxima de corriente que puede consumir. El microcontrolador ESP32 tiene una capacidad máxima de 1200mA, aunque el propio dispositivo ya consumirá un máximo de 180mA.

El módulo relé de 2 canales utiliza una señal de control para manejar la activación de las bobinas, debido a que su activación es por nivel bajo, cuando las 2 bobinas están apagadas el consumo será de 20mA (10mA por cada relé). La activación de los relés se utilizará para el movimiento ascendente/descendente de la persiana, por lo que la implementación del microcontrolador no permitirá la activación de los 2 relés al mismo tiempo. Sabiendo esto, el máximo consumo del módulo relé será cuando un relé está activado (90mA de la bobina) y el otro relé desactivado (10mA de la señal de control), lo que supone un total de 100mA.

El módulo RTC DS3231 tiene un consumo en reposo de 170 μ A, pero cuando el microcontrolador se comunica con el reloj a través de la interfaz I²C el consumo asciende a 300 μ A (0,3mA).

También es necesario contabilizar el consumo de corriente de los botones y diodos led que tendrá conectado el microcontrolador. El ESP32 tiene un consumo máximo de 12mA por cada pin de E/S de la placa, por lo que el uso de 3 botones y 2 diodos led no puede superar los 60mA.

TABLA 4.5. CONSUMO DE CORRIENTE DEL DCP

Componente	Consumo de corriente máximo
ESP32	180mA
Módulo relé	100mA
DS3231	0,3mA
Pines E/S ESP32	60mA
Total	340,3mA

Fuente: Elaboración propia

Con un consumo total del DCP de 340,3mA, el ESP32 será capaz de alimentar sin problema todos los componentes del circuito, evitando así el uso de una alimentación de corriente auxiliar que incrementaría el coste del proyecto. El dispositivo DCP deberá ser alimentado por su conector Micro-USB y con un amperaje que supere el consumo calculado.

4.6. Diseño de la aplicación.

La aplicación móvil es el componente software que permitirá el control y configuración de los DCP que se encuentren en la misma red Wi-Fi. El diseño de la aplicación incluye una propuesta de interfaz a modo de prototipo, que servirá para enfocar el diseño gráfico para su posterior implementación. Además, se tratará el funcionamiento lógico y comunicaciones con los DCP, así como las decisiones de diseño tomadas.

4.6.1. Sistema operativo y entorno de desarrollo.

En la actualidad existen dos principales mercados móviles donde lanzar una aplicación. En primer lugar, los iPhone que utilizan el sistema operativo IOS y la segunda opción, los dispositivos con sistema operativo Android.

Android tiene una clara ventaja frente a IOS, y es que la gran variedad de dispositivos Android existentes y el bajo precio de su gama de entrada le han permitido hacerse con un gran porcentaje de la cuota de mercado. Según la web Statcounter, en mayo de 2020 a nivel mundial Android ocupaba un 72,52% de la cuota de mercado, mientras IOS tenía un 26,8%, sumando entre los dos más de un 99% del total de dispositivos. [15]

Existen otros factores que decantan la balanza a favor de un desarrollo en Android, como es la accesibilidad a dispositivos Android para poder realizar pruebas durante la fase de implementación. Además, se debe tener en cuenta el tiempo disponible para realizar la implementación de la aplicación y los conocimientos en lenguaje Java del desarrollador permitirán que el tiempo de adaptación al entorno de desarrollo Android Studio sea mucho menor que el aprendizaje de un lenguaje y entorno totalmente nuevos.

Por todos estos motivos se ha decidido implementar la aplicación móvil en el sistema operativo Android, en el entorno de desarrollo Android Studio y con el lenguaje de programación Java. A pesar de esto, a continuación, se presentan algunas alternativas posibles.

En un desarrollo para el sistema operativo IOS, podría utilizarse el lenguaje de programación Swift con el framework Cocoa. Swift se trata del lenguaje de programación creado por Apple para el desarrollo de aplicaciones en sus dispositivos.

Angular se trata de un framework desarrollado por Google que tiene como principal lenguaje de programación Typescript. Tiene como principal fortaleza un desarrollo rápido y fácil de las aplicaciones.

4.6.2. Prototipo de interfaz de usuario.

La realización de prototipos de la interfaz de usuario permite obtener una idea inicial del resultado visual del proyecto, en el que reflejar de manera rápida y sencilla los elementos necesarios de la aplicación. La creación de Mock-Ups de las principales pantallas de la APP ha permitido conseguir un prototipo de interfaz de usuario con un bajo coste temporal que refleja de manera sencilla aquella funcionalidad e información que se quiere implementar en la aplicación final.

La pantalla principal (figura 4.4.) debe mostrar el listado de todos los DCP conectados (caso de uso CU-03), así como los grupos que se hayan creado (caso de uso CU-04). De forma intuitiva se podrá acceder a la configuración de cada dispositivo seleccionando el elemento correspondiente de la lista, lo que accederá a la pantalla de control (figura 4.5.). Además, existe una barra de herramientas con los botones de refrescar y añadir. El botón de refrescar recargará toda la información de la APP, mientras que con el botón de añadir se podrán crear grupos o añadir un nuevo DCP.

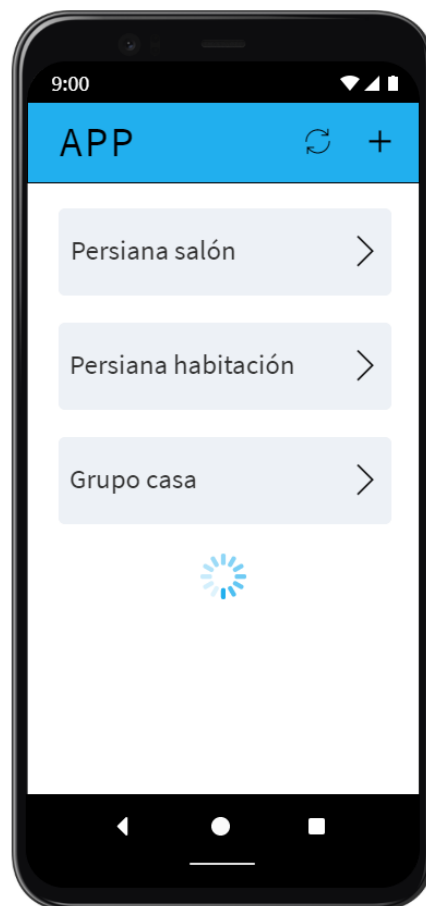


Fig. 4.6. Mock-Up Pantalla principal. Fuente: Elaboración propia

La pantalla de control (figura 4.5.) muestra la configuración y estado de un dispositivo DCP o grupo. En primer lugar, se muestra el nombre asociado al DCP o grupo, en este ejemplo se trata del dispositivo asociado a la persiana del salón. Junto al nombre existe un botón de editar que permitirá modificar el mismo. El siguiente elemento es una barra de progreso, la cual nos permitirá modificar la altura de la persiana deslizando y soltando en la altura deseada (caso de uso CU-07). En la barra de herramientas se muestra un botón con el icono de una basura, que permitirá borrar un grupo seleccionado (caso de uso CU-09).

El resto de los elementos mostrados se corresponden con la configuración de la programación horaria del dispositivo. Un pequeño interruptor o “switch” nos permite activar o desactivar la programación horaria completa del dispositivo sin necesidad de borrar la configuración actual establecida. A continuación, se mostrarán todos los desplazamientos configurados en forma de lista. En el ejemplo mostrado existe un único movimiento que registra un desplazamiento, abriendo por completo la persiana (100% altura) a las 9:00 y que solo tendrá efecto en los días remarcados (de lunes a viernes). Al seleccionar uno de los desplazamientos programados o al pulsar el botón de añadir, se accederá a la pantalla de edición de programación horaria (figura 4.6), permitiendo modificar o crear desplazamientos correspondientemente.

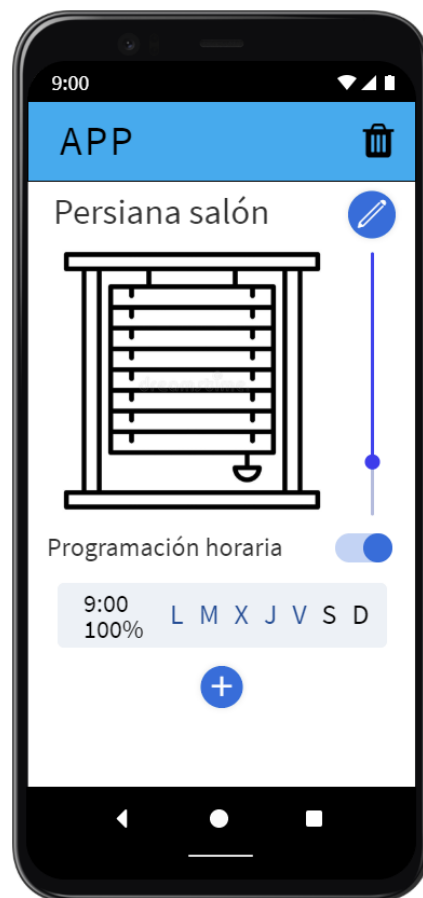


Fig. 4.7. Mock-Up Pantalla de control. Fuente: Elaboración propia

La pantalla de edición de la programación horaria (figura 4.6.) permite tanto modificar como crear desplazamientos automáticos en un DCP o grupo (CU-09). En primer lugar, se debe seleccionar la altura de la persiana a la que se debe desplazar, utilizando la barra de progreso. Pulsando en un día de la semana se puede activar (remarcados en azul) o desactivar (en negro), lo que permite aplicar el movimiento solo a los días seleccionados. Utilizando el reloj mostrado y haciendo scroll, se puede seleccionar la hora sobre la que realizar el desplazamiento. Por último, se muestran dos botones, uno con el cual eliminar la configuración actual y otro para guardar las modificaciones.

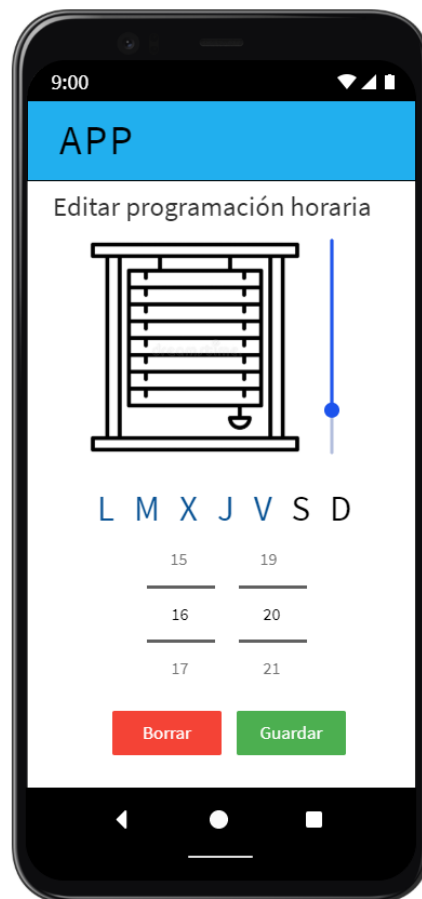


Fig. 4.8. Mock-Up Pantalla edición de programación horaria. Fuente: Elaboración propia

Al crear un nuevo grupo (caso de uso CU-05), aparecerá un Pop-Up (figura 4.7.) que permitirá seleccionar el nombre inicial del grupo. Tras darle al botón de “Crear” se cerrará el Pop-Up y aparecerá un nuevo grupo en la pantalla principal de la aplicación.

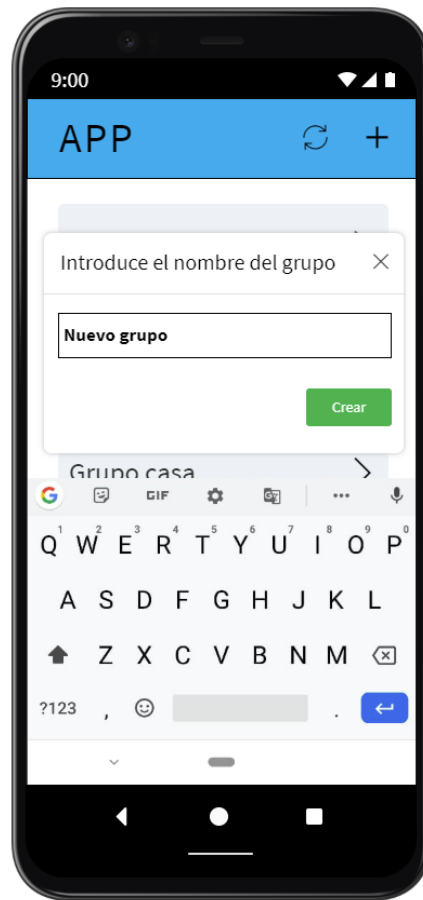


Fig. 4.9. Mock-Up Pantalla de creación de grupo. Fuente: Elaboración propia

Finalmente, al entrar en el modo de sincronizar desplazamiento de una persiana (caso de uso CU-06), aparecerá la pantalla de sincronización (figura 4.8.). Se pueden ver dos botones que se deberán mantener pulsados para subir y bajar la persiana por completo para efectuar correctamente la sincronización. En la barra de herramientas existe un botón de ayuda que indica las instrucciones para realizar la sincronización.

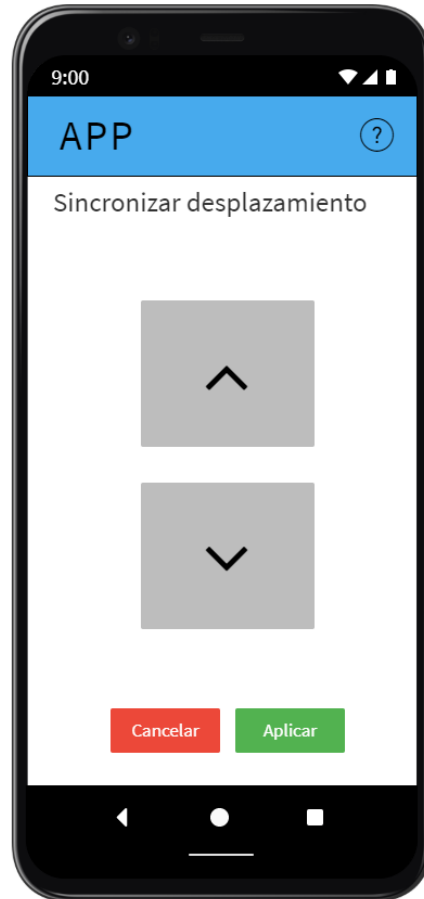


Fig. 4.10. Mock-Up Pantalla de sincronización de desplazamiento. Fuente: Elaboración propia

5. IMPLEMENTACIÓN

En esta sección se desarrolla y detalla toda aquella implementación y codificación realizada para alcanzar un óptimo funcionamiento del proyecto.

5.1. Implementación del DCP.

Para la implementación del dispositivo de control de persianas se ha utilizado el microcontrolador ESP32, según se argumenta en la sección de diseño. La codificación de este microcontrolador se puede realizar con el lenguaje de programación de Arduino y además cuenta con una total compatibilidad con el entorno de desarrollo de Arduino (Arduino IDE).

El lenguaje de programación de Arduino está basado en C++ y soporta gran parte de los comandos de C y C++. [16]

Para permitir el uso de la plataforma de Arduino con el ESP32, la compañía Espressif Systems pone a disposición un repositorio público en Git, que se debe clonar en el directorio de instalación del IDE de Arduino. [17]

Debido a la simplicidad del IDE de Arduino, se ha optado por utilizar Visual Studio Code como entorno de desarrollo para la implementación del microcontrolador. Visual Studio Code es una herramienta desarrollada por Microsoft, cuya principal finalidad es la edición de código fuente. La principal ventaja de utilizar este software es la posibilidad de utilizar Git, con el que tener un control de versiones fiable y en la nube. Además, está disponible de la extensión “Arduino”, que integra en Visual Studio Code la funcionalidad de la plataforma de Arduino.

5.1.1. Esquema eléctrico.

A la hora de realizar una implementación con un microcontrolador, es de gran ayuda utilizar un esquema con la disposición de los pines que ofrece la placa que estemos utilizando. La siguiente imagen muestra la disposición de los pines de la placa ESP32 utilizada.

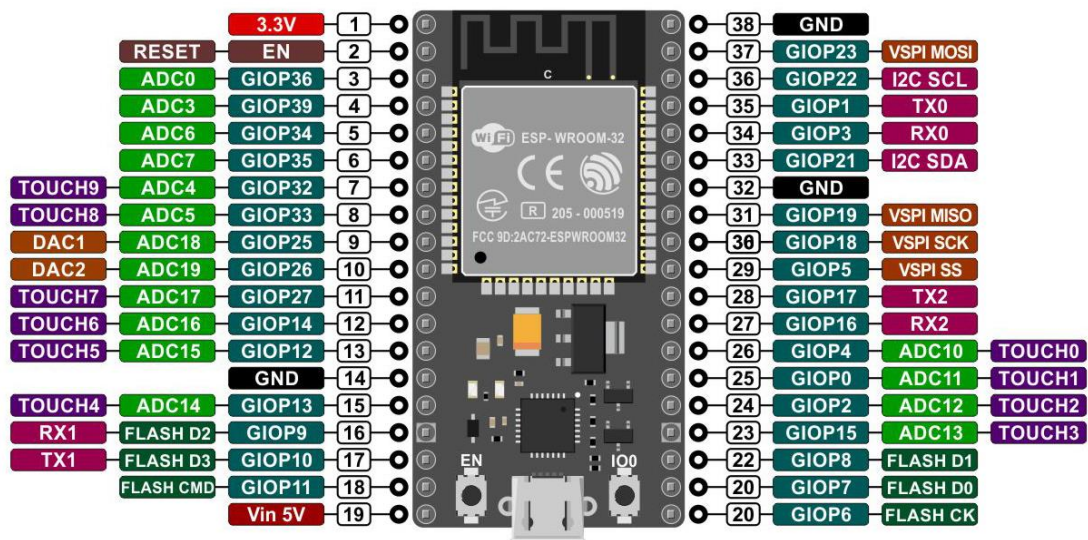


Fig. 5.1. Disposición de los pines del ESP32. Fuente: [18]

El circuito implementado para el desarrollo del DCP gira en torno al microcontrolador ESP32 sobre el cual se conectarán el resto de los componentes. Existen 3 botones que permiten el control manual, el botón de subir, el de bajar y el botón de función, que están conectados con los pines 25, 26 y 27 respectivamente del microcontrolador. Los leds de estado del DCP están conectados a los pines 12 (indicador conexión Wi-Fi) y 13 (indicador modo de funcionamiento), utilizando una resistencia de 220Ω para cada uno de ellos. El módulo relé de doble canal se conectará desde su pin VCC al pin de 5 voltios de microcontrolador suministrando corriente a las bobinas. Además, los pines IN1 e IN2 sirven como señal para activar y desactivar cada relé, conectándose con los pines 32 y 33 del microcontrolador. El módulo RTC se conectará al pin de 3,3 voltios del microcontrolador, y como transmite la información mediante la interfaz I²C se conectarán los pines SDA y SCL con los pines 21 y 22 respectivamente. Por último, todos los componentes auxiliares se deben conectar a un pin de tierra GND del microcontrolador.

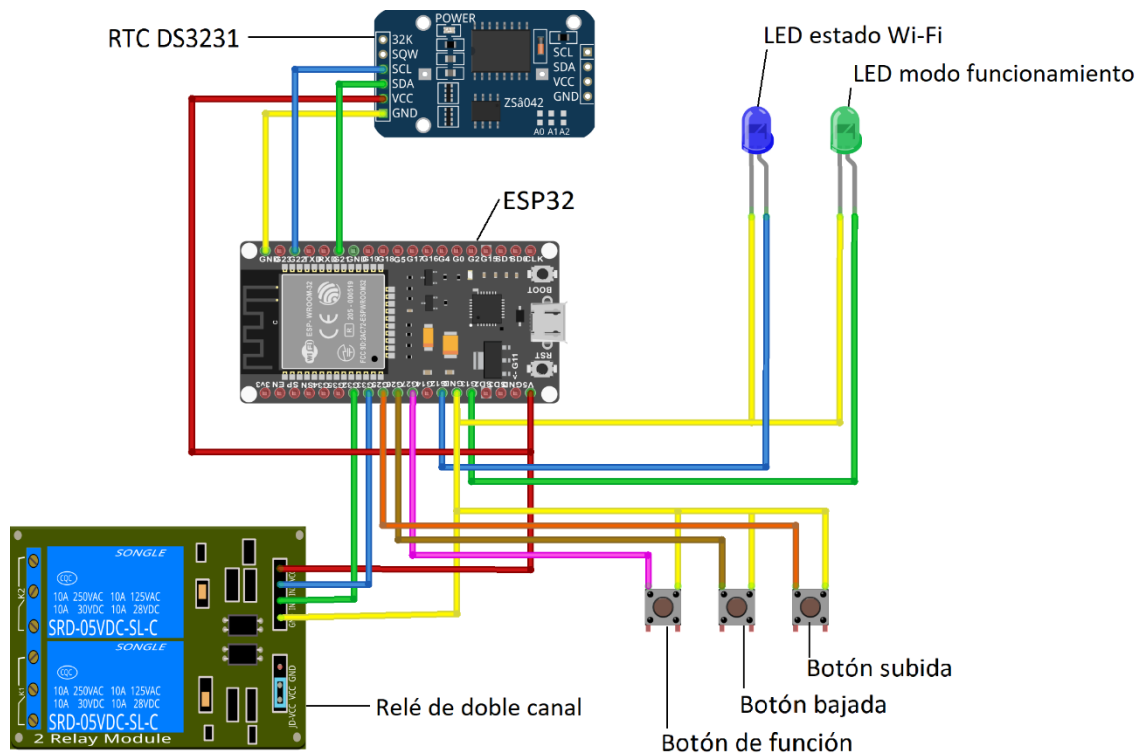


Fig. 5.2. Diseño del circuito del DCP. Fuente: Elaboración propia

5.1.2. Codificación.

A continuación, se realiza una breve explicación de la codificación implementada en el microcontrolador, haciendo énfasis en las principales dificultades y su resolución.

La funcionalidad principal del proyecto consiste en el control y automatización de una persiana eléctrica, y por ello se comenzó con la programación de la activación de los relés de subida y bajada del motor eléctrico.

Para ello se han utilizado dos variables que representan la posición actual de la persiana y la posición que se debe alcanzar de la persiana, ambas representadas como un porcentaje, por lo que tomarán valores del 0 (persiana totalmente bajada) al 100 (persiana totalmente subida). La variable que almacena la posición actual de la persiana se irá actualizando según se realice el movimiento, incrementándose o decrementándose en una unidad, y activando el relé correspondiente el tiempo requerido para el movimiento de un 1% del desplazamiento total de la persiana. La variable que almacena la posición que debe alcanzar la persiana cambiará de valor en función de las acciones del usuario, desde la aplicación móvil o con los botones físicos de movimiento del DCP. Cuando la variable de posición actual y la de posición a alcanzar toman el mismo valor la persiana se para, en caso de tener valores diferentes se activará el relé correspondiente para el movimiento de la persiana hasta que se consigan igualar dichos valores.

Por sencillez y para abaratar costes, se ha optado por no utilizar ningún sensor que permita al microcontrolador saber la posición actual de la persiana. Por este motivo, el desplazamiento de la persiana se realiza en periodos de tiempo que corresponden con el 1% del desplazamiento total. Se debe tener en cuenta que cada persiana tendrá tiempos diferentes en su desplazamiento, incluso para una misma persiana, el tiempo de subida y bajada puede diferir un poco debido a que el trabajo realizado no es el mismo. Este problema se soluciona con una fase de sincronización realizada desde la aplicación móvil, donde el usuario tendrá que bajar y subir por completo la persiana y el microcontrolador se encargará de almacenar el tiempo efectuado para cada desplazamiento.

La funcionalidad del movimiento de la persiana mediante una programación horaria se podrá activar y desactivar mediante un botón físico en el DCP o mediante la aplicación móvil.

El microcontrolador almacena la programación horaria en un array bidimensional, donde cada posición almacena la hora y el minuto cuando se realizará el movimiento y la nueva posición que debe alcanzar la persiana. El array tiene 7 filas, cada una de ellas corresponde con un día de la semana y las columnas almacenan cada uno de los movimientos para un día de la semana determinado. Cuando la funcionalidad de programación horaria se encuentra activada, el microcontrolador obtiene la fecha y hora actual del módulo RTC DS3231 y al inicio de cada minuto comprobará si existe algún movimiento almacenado en el array para ese día-hora-minuto. En caso de que exista un desplazamiento para el momento actual, se actualizará la variable de posición a alcanzar por la persiana con la establecida en la programación horaria.

La existencia de botones físicos permite controlar la funcionalidad básica del DCP, pero para un control más cómodo y para tener acceso a opciones más avanzadas se requiere el uso de un dispositivo móvil, que se comunicará con el DCP a través de una conexión TCP/IP. Para que esto sea posible se ha utilizado la librería de Arduino “Wi-Fi.h” que nos permite atender peticiones TCP en un puerto determinado, a modo de servidor. Cuando se abre una nueva conexión TCP se recibe y almacena la información enviada por el cliente (en este caso la aplicación móvil). La información transmitida debe tener el siguiente formato < RQ-XXX;DT-Y; >, donde XXX es un valor numérico que representa el identificador de una orden y la Y es una cadena de caracteres de tamaño indefinido que tendrá información adicional para la realización de la orden en caso de ser necesario. Por ejemplo, < RQ-001;DT-075; > será la información que se debe enviar al DCP para mover la persiana hasta un 75% de su altura total, el identificador “001” se refiere a la orden de desplazar la persiana y “075” es el nuevo valor de altura al que se debe desplazar la persiana. Las órdenes recibidas por el DCP deben seguir la estructura especificada para ser tratadas, además, cada una de las diferentes ordenes puede tener verificaciones adicionales para comprobar que los datos adicionales cumplen las condiciones necesarias.

En la tabla 5.1. se identifican cada una de las ordenes que puede atender el DCP.

Una vez identificada y verificada la orden recibida el microcontrolador intentará realizar la acción, responderá al emisor de la orden mediante la conexión TCP y por último cerrará la conexión.

Es importante destacar que el microcontrolador no soportará conexiones TCP simultaneas, por lo que se ha optado por establecer un tiempo máximo en la conexión o timeout, que finalizará la transmisión de información una vez transcurrido este periodo (10 segundos).

TABLA 5.1. ÓRDENES PERMITIDAS DEL DCP

Orden	Identificador	Información adicional
Mover persiana	001	Número entero entre 0 y 100 que indica la posición a la que se debe desplazar la persiana.
Ajustar tiempo de ascenso	002	Número natural que indica el tiempo de un ascenso total de la persiana. Valor en milisegundos.
Ajustar tiempo de descenso	003	Número natural que indica el tiempo de un descenso total de la persiana. Valor en milisegundos.
Ajustar fecha y hora	004	Nueva fecha y hora que utilizará el DCP. Debe tener el formato YYYY/MM/DD-hh:mm:ss.
Actualizar programación horaria	006	Nombre del grupo a modificar. Cadena de texto con toda la programación horaria del DCP. Se utiliza el formato hhmm%%%, donde “hh” son las horas, “mm” los minutos y “%%%” el porcentaje de altura al que se debe desplazar la persiana.
Borrar programación horaria	007	Nombre del grupo sobre el que borrar la programación horaria.
Cambio de modo (“solo manual” o “manual + programación horaria”)	009	Cadena de texto: “active” activa la programación horaria y “inactive” la desactiva.
Cambiar nombre de un grupo	011	Nombre actual y nuevo nombre, separados por el carácter ‘:’.
Enviar estado actual	012	Sin datos adicionales.
Crear nuevo grupo	013	Nombre del nuevo grupo.
Borrar grupo	014	Nombre de un grupo existente.

Enviar programación horaria	015	Nombre de un grupo existente.
-----------------------------	-----	-------------------------------

Fuente: Elaboración propia

La comunicación TCP funciona como un modelo cliente-servidor, donde el usuario desde la aplicación móvil envía peticiones al DCP para realizar una determinada acción, y el DCP actúa como servidor atendiendo las peticiones de la aplicación móvil y respondiendo con información o con el resultado de la acción realizada. El problema reside en que en un primer momento la aplicación móvil no conoce la dirección IP de los DCP, ya sea porque nunca ha tenido contacto con ellos, o porque su dirección IP ha podido cambiar. Como solución se ha optado por la difusión de paquetes UDP tipo broadcast que identifiquen al DCP en toda la red local de manera periódica, permitiendo conocer al dispositivo móvil la información básica del DCP y su dirección IP.

El DCP debe ser un dispositivo tolerante a cortes de energía, lo que supuso la incorporación del módulo RTC DS3231 que permitía mantener la fecha y hora, pero este no es el único dato que debe persistir tras recuperar la corriente eléctrica. Existen datos del programa, como la posición actual de la persiana o la configuración de la programación horaria que no se pueden perder, ya que supondría una reconfiguración por parte del usuario cada vez que se produzca un corte de energía. Para solventar este problema se ha utilizado la librería “EEPROM.h” que permite utilizar hasta 4MB de la memoria flash del microcontrolador para almacenar datos de manera persistente. Cada vez que una de estas variables globales se actualiza, también se actualizará su valor en la memoria flash del dispositivo. De esta manera cuando el microcontrolador se enciende y ejecuta la función setup(), los valores de dichas variables se cargan desde la memoria flash permitiendo continuar un correcto funcionamiento de manera transparente para el usuario.

El DCP pone a disposición del usuario la opción de hacer un reset del dispositivo, lo que borrará toda la configuración almacenada. Para implementar esta funcionalidad se ha creado una función que borra por completo todos los valores guardados en la memoria flash del microcontrolador y almacenará valores por defecto para las variables que lo necesiten. También es necesario borrar la configuración Wi-Fi (SSID y contraseña) almacenados. Por último, se ejecutará la función ESP.restart() (específica del ESP32) que permite hacer un reinicio del dispositivo.

El microcontrolador ESP32 es capaz de almacenar las credenciales de una red Wi-Fi (SSID y contraseña) en su memoria flash, sin embargo, necesita recibir estos datos para establecer la conexión por primera vez. La opción fácil y rápida para establecer una conexión Wi-Fi es añadir las credenciales en forma de hardcode. Sin embargo, se pretende que el dispositivo sea versátil y se pueda conectar a una nueva red Wi-Fi de la manera más sencilla para el usuario. Por este motivo se ha utilizado la funcionalidad “smartConfig” que ofrece el ESP32, que crea un punto de acceso Wi-Fi a través del cual otro dispositivo, como por ejemplo un smartphone, puede conectarse y transmitir las

credenciales Wi-Fi de forma segura. Cuando el DCP se enciende por primera vez o después de un reseteo se tendrá que configurar la red Wi-Fi y tras este proceso el dispositivo se conectará de forma automática siempre que la red esté disponible.

El microcontrolador seleccionado es el ESP32 y como se especifica en la sección de diseño, se posee un procesador de doble núcleo. Haciendo uso de variables del tipo `TaskHandle_t` y la función `xTaskCreatePinnedToCore()` se pueden crear nuevos procesos de ejecución, especificando incluso en cuál de los núcleos se debe ejecutar.

Por defecto, las funciones `setup()` y `loop()` se ejecutan en el núcleo número 1, esto es fácilmente comprobable haciendo una llamada a la función `xPortGetCoreID()` la cual devuelve el valor del núcleo que ejecuta la llamada. Sabiendo esto se puede repartir la carga de trabajo entre los dos núcleos y además utilizar la gestión de procesos, se puede priorizar la realización de ciertas tareas.

En nuestro caso, se ha dejado el núcleo 1 exclusivamente para la ejecución de la función `loop()`, la cual se ejecutará constantemente realizando tareas prioritarias y que no deben esperar para su ejecución. En este núcleo se ejecutarán las tareas de lectura de los botones físicos, se encarga de cambiar el modo de funcionamiento del dispositivo y hacer el reset cuando así se requiera, comprueba la configuración de programación horaria y actualiza la variable de posición a alcanzar con el nuevo valor, se realiza el salvado de las variables globales en la memoria flash cuando son modificadas y, por último, controla el encendido y apagado de los leds de estado del DCP.

La función `setup()` se ejecuta al encender el microcontrolador y, a parte de la inicialización de variables, pines de lectura o escritura, la memoria flash y el módulo RTC DS3231, también se encarga de la creación de dos procesos que se ejecutarán en el núcleo 0. En primer lugar, la función `adjustHeight()` la cual se encarga de la activación de los relés de bajada y subida de la persiana cuando se requiere. En segundo lugar, la función `startTCPConnection()` atiende las peticiones TCP entrantes cuando la conexión Wi-Fi está activada. Cuando llega una nueva petición TCP y finaliza el envío de datos del emisor, se trata y se ejecuta la orden solicitada (si cumple con los requisitos definidos), después se responde y se cierra la conexión TCP.

Uno de los inconvenientes de la conexión TCP es que, con el código utilizado, se permanece en una misma sección de código hasta que finaliza la transmisión de datos y se cierra la conexión. Si todas las tareas se ejecutasen en el mismo proceso, esto provocaría que el resto de las tareas prioritarias, como lectura de botones físicos o control de leds de estado quedasen totalmente bloqueadas. Al utilizar un proceso separado para el tratamiento de las conexiones TCP se evita de una manera sencilla que este problema pueda ocurrir.

5.2. Implementación de la aplicación.

La aplicación se desarrollará para el sistema operativo Android, con el entorno de desarrollo Android Studio y bajo la API 24 correspondiente a la versión Android 7.0 Nougat. En este momento, junio del 2020, la API 24 está soportada un 73,7% de todos los dispositivos Android. En la figura 5.2. se muestra la distribución de los dispositivos compatibles por versión y API de Android.

4.0	Ice Cream Sandwich	15	
4.1	Jelly Bean	16	99,8%
4.2	Jelly Bean	17	99,2%
4.3	Jelly Bean	18	98,4%
4.4	KitKat	19	98,1%
5.0	Lollipop	21	94,1%
5.1	Lollipop	22	92,3%
6.0	Marshmallow	23	84,9%
7.0	Nougat	24	73,7%
7.1	Nougat	25	66,2%
8.0	Oreo	26	60,8%
8.1	Oreo	27	53,5%
9.0	Pie	28	39,5%
10.	Android 10	29	8,2%

Fig. 5.3. Distribución de dispositivos compatibles por API de Android. Fuente: Android Studio

5.2.1. Codificación.

La creación de aplicaciones en Android Studio se compone principalmente de las clases Java que reflejan la parte lógica de la aplicación, mientras el aspecto visual reside en archivos XML. A continuación, se va a realizar una breve explicación de las principales clases desarrolladas en el proyecto Android.

En primer lugar, se ha creado una clase “Constants” donde almacenar valores constantes, como puede ser el valor del puerto del socket para las comunicaciones, valores por defecto o el número de cada petición del DCP mostrada en la tabla 5.1.

También se ha implementado una clase “ConnectionTCP” que contiene métodos estáticos y públicos que facilitarán el envío de peticiones al DCP en el resto del código de la aplicación. El primero de los métodos, llamado “request”, recibe por parámetro la dirección IP del destinatario, el ID de la petición a realizar y los datos extra de la petición. El método establecerá una conexión TCP con el dispositivo de destino y enviará los datos de la petición con el formato esperado por el DCP. El método esperará una respuesta del DCP que será retornada en forma de String tras cerrar la conexión TCP. El segundo método “requestAsync”, funciona igual que “request” con la diferencia que se realiza de forma asíncrona en un proceso separado y sin retornar un valor String con la respuesta del DCP.

Se han creado las clases DCP y GroupDCP para almacenar los datos recibidos por Wi-Fi de los dispositivos. DCP contiene los atributos principales de un dispositivo: el nombre, la dirección IP, el porcentaje de altura actual y el modo de programación horaria. GroupDCP almacena el nombre del grupo y una lista del tipo ArrayList con las direcciones IP de los dispositivos pertenecientes al grupo.

La pantalla principal de la aplicación está gestionada por la clase MainActivity cuya principal tarea es recibir los datos de los DCP mediante un socket UDP. Cuando se reciben datos de un DCP por primera vez se crea su correspondiente instancia y si se trata de un DCP ya almacenado, se actualizarán los datos si existen cambios. La clase MainActivity se encarga de mostrar los datos de los distintos DCP y grupos en forma de lista, haciendo uso del componente RecyclerView. Además, para facilitar la usabilidad de la aplicación, se han implementado diferentes pestañas o tabs, una para la visualización de los DCP y otra para los grupos.

Al seleccionar un DCP o grupo del listado de la pantalla principal se accederá a una pantalla donde controlar y configurar los dispositivos. La lógica del control de un DCP se encuentra en la clase ControlDCPActivity, que dispone de las funcionalidades básicas de control del DCP. Cuando se realiza una acción en esta pantalla, los datos son enviados al DCP mediante una conexión TCP. Poniendo un ejemplo, si se solicita mover la persiana hasta el 50% de altura, la aplicación enviará los siguientes datos al DCP <RQ-001;DT-050;> donde 001 se corresponde con “mover persiana” y 050 es el valor al que debe desplazarse (50%). En el caso de los grupos, la lógica reside en ControlGroupsActivity, que es muy similar a ControlDCPActivity y cuya principal diferencia se encuentra en que, en el caso de los grupos, los datos deben ser enviados a todos y cada uno de los dispositivos pertenecientes al grupo, estableciendo varias conexiones TCP simultáneas.

El DCP utiliza la funcionalidad “smartConfig” para recibir las credenciales Wi-Fi y debe ser en aplicación donde se implemente la lógica para conectarse al punto de acceso del ESP32, cifrar los datos y enviarlos. Para poder incluir dicha funcionalidad se ha reutilizado el código que pone a disposición Espressif Systems. A través de GitHub se ha obtenido el proyecto Android de la aplicación propuesta por Espressif Systems y tras un análisis general del funcionamiento se ha adaptado dicha implementación en este

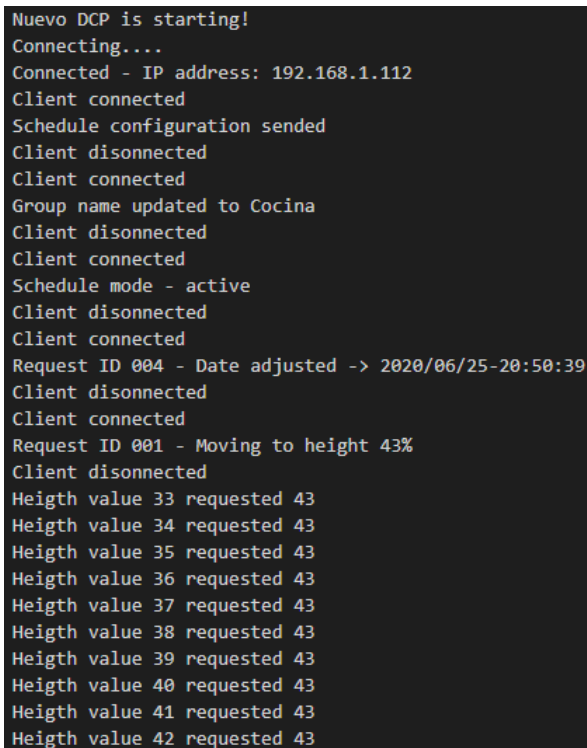
proyecto permitiendo la transmisión de los datos Wi-Fi de forma segura y fácil para el usuario.

5.3. Pruebas.

La realización de pruebas es una parte fundamental de un proyecto que permite encontrar fallos de funcionamiento y la búsqueda exhaustiva de casos problemáticos permite sacar a la luz estos fallos. Este proyecto está dividido en el desarrollo del DCP y el de la aplicación, y según se ha ido implementando funcionalidad se ha comprobado que todos los procesos anteriores no se hayan visto afectados. Esta rutina de verificación de la funcionalidad ha permitido que la fase de pruebas final haya durado muy poco tiempo.

5.3.1. Pruebas del DCP.

Las pruebas del DCP están principalmente ligadas a la funcionalidad que debe prestar, a su vez, el servicio de peticiones listado en la tabla 5.1. Como el componente se debe probar por separado para verificar el correcto funcionamiento antes de su integración con el software de control, el envío de peticiones Wi-Fi se ha realizado mediante un pequeño programa Java. La principal forma utilizada para comprobar el funcionamiento del DCP ha sido el Monitor Serial, que es la salida por pantalla del microcontrolador.



```
Nuevo DCP is starting!
Connecting...
Connected - IP address: 192.168.1.112
Client connected
Schedule configuration sended
Client disconnected
Client connected
Group name updated to Cocina
Client disconnected
Client connected
Schedule mode - active
Client disconnected
Client connected
Request ID 004 - Date adjusted -> 2020/06/25-20:50:39
Client disconnected
Client connected
Request ID 001 - Moving to height 43%
Client disconnected
Height value 33 requested 43
Height value 34 requested 43
Height value 35 requested 43
Height value 36 requested 43
Height value 37 requested 43
Height value 38 requested 43
Height value 39 requested 43
Height value 40 requested 43
Height value 41 requested 43
Height value 42 requested 43
```

Fig. 5.4. Captura de pantalla del Monitor Serial. Fuente: Elaboración propia

Durante esta fase del proyecto no se han realizado pruebas con ningún motor de persiana, pero se han aprovechado los leds que dispone el módulo relé para verificar que las activaciones del motor son las correctas.

El resultado de las pruebas ha sido satisfactorio, mostrándose en todo momento el resultado esperado basándonos en los requisitos.

5.3.2. Pruebas de la aplicación.

Desde el inicio del desarrollo de la aplicación, ya se contaba con el DCP implementado, lo que facilitaba las pruebas de funcionamiento. En este caso la realización de pruebas es mucho menor y más sencilla ya que no es mucha la lógica detrás de la misma. La aplicación principalmente se limita a mostrar los dispositivos de control que se encuentran en la red y enviar peticiones TCP con un formato ya preestablecido. La gestión y el listado de la información de los DCP y grupos ha sido el mayor reto en cuanto a pruebas y corrección de errores.

Todas las pruebas finales de funcionamiento se han comprobado con un mínimo de tres DCP conectados en la red, y se han realizado pruebas de movimiento manual, de movimiento automático con el planificador, y de cada una de las funcionalidades de la interfaz para garantizar que los botones cumplen con los requisitos.

5.3.3. Pruebas conjuntas.

Para llevar a cabo una comprobación de la funcionalidad final del proyecto, se ha utilizado un DCP conectado a una persiana motorizada, pudiendo ver en funcionamiento la idea inicial del proyecto. Tras la instalación del motor de persiana y la conexión eléctrica con el DCP, se ha utilizado la aplicación para probar, una a una, todas las funcionalidades del proyecto plasmadas en los requisitos funcionales en la sección de “Diseño” y comprobando el cumplimiento de los criterios de aceptación de cada requisito.

El resultado de las pruebas de integración ha sido un éxito. Todas las funcionalidades se han realizado sin errores y sin necesidad de cualquier acción inesperada, o adaptación. Las pruebas se han repetido cinco veces para verificar la estabilidad del sistema, incluyendo pruebas de desconexiones para simular errores, que han sido correctamente tratados.

6. RESULTADOS

En esta sección se realizará un análisis de los resultados obtenidos, explicando y detallando toda la funcionalidad implementada y como usarla. Tal como se ha hecho hasta ahora, se diferenciará un apartado para los resultados del DCP y otro para la aplicación. Además, se realizará un análisis global de los resultados aportados por el proyecto.

6.1. Resultados del DCP.

El componente físico diseñado e implementado, denominado como DCP (dispositivo de control de persiana), se ha conseguido gracias a un estudio de los componentes necesarios y el desarrollo de su software. El circuito mostrado en la figura 6.1. es un ejemplo real de implementación del dispositivo en una protoboard.

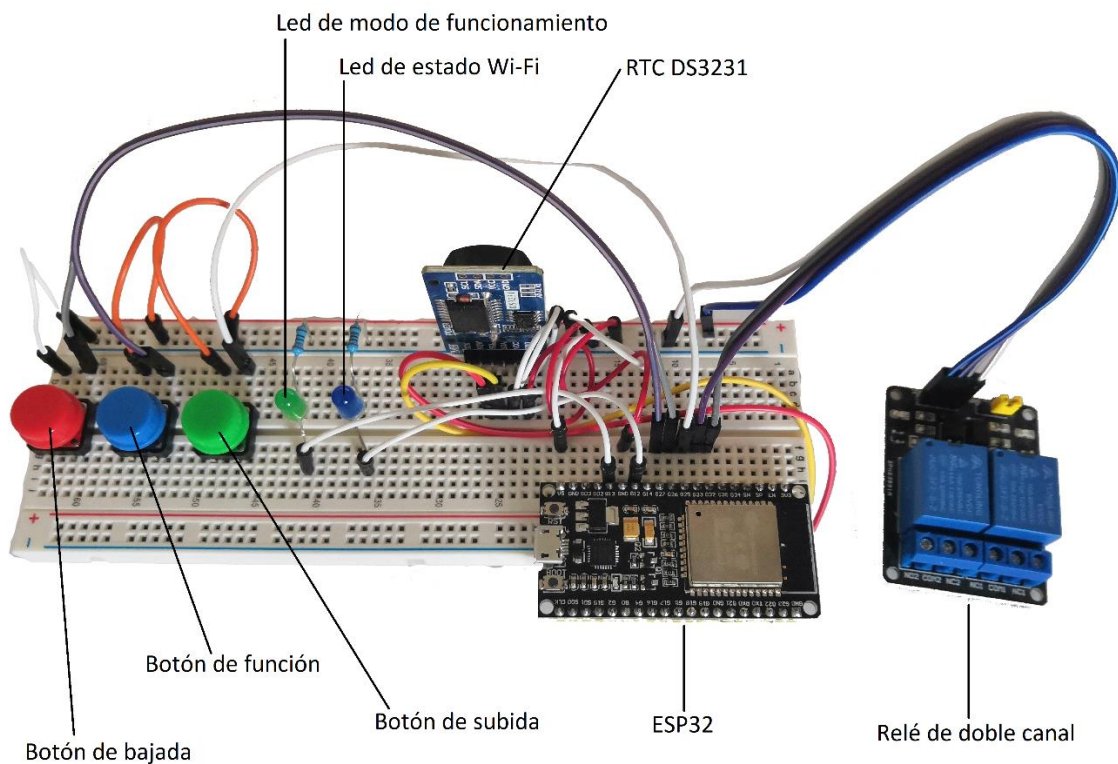


Fig. 6.1. Circuito del DCP en una protoboard. Fuente: Elaboración propia

Una vez se tiene el circuito, el motor de persiana se debe conectar a la corriente del hogar y al DCP. El proyecto está pensado y definido para poder controlar cualquier motor de persiana que posea las conexiones estándar; estas son: un cable para la toma de tierra, un cable para la conexión neutra del enchufe y dos cables para la conexión de

fase (uno para subir y otro para bajar). Tanto la toma de tierra como el neutro se deben conectar de manera normal a la instalación eléctrica del hogar y serán los dos cables de subida/baja los que se conectarán al a fase mediante el módulo relé, como se muestra en la figura 6.2.

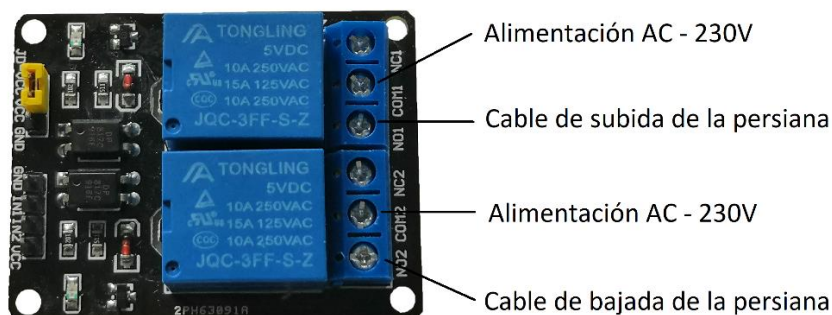


Fig. 6.2. Conexión eléctrica con el módulo relé. Fuente: Elaboración propia

Una vez se ha realizado la instalación del DCP, la funcionalidad que ofrece únicamente con el control físico se limita a sus tres botones. Los botones de subida y bajada sirven para activar el motor, permitiendo el desplazamiento de la persiana en ambas direcciones mientras se mantenga pulsado el botón. Además, se pueden combinar cualquiera de estos dos botones con el botón de función para que se realice un desplazamiento completo de la persiana sin mantener pulsado los botones de desplazamiento. Por ejemplo, si la persiana está bajada y se quiere subir por completo, el usuario podría mantener pulsado el botón de subir hasta alcanzar el límite superior, pero para mayor comodidad, si se pulsa el botón de función y seguidamente un click en el de subir, la persiana se desplaza automáticamente hasta arriba. Además, si se presiona dos veces seguidas el botón de función, se cambiará el modo de funcionamiento del DCP, alternando entre programación horaria activada y desactivada. Por último, si se mantiene pulsado durante el botón de función durante 3 segundos el DCP borrará toda su configuración, eliminando credenciales Wi-Fi, programación horaria, etc....

El DCP cuenta con dos diodos led que ofrecen al usuario información básica sobre el estado del dispositivo. Como se ve en la figura 6.1. el led azul se corresponde con el estado de la conexión Wi-Fi, cuando la no se ha establecido ninguna conexión o se ha perdido la red configurada, el led se iluminará de forma intermitente, y cuando se establece una conexión Wi-Fi correctamente el led permanece iluminado. El led verde se corresponde con el modo de funcionamiento del DCP, iluminándose cuando la programación horaria está activada y apagándose cuando esté desactivada.

6.2. Resultados de la aplicación.

La aplicación se ha diseñado con el fin de permitir una configuración y comunicación rápida con el DCP implementado una interfaz sencilla, basándose en el prototipo creado en la fase de diseño. A continuación, se muestran una serie de capturas de pantallas de la aplicación y una breve explicación de cada una, mostrando un ejemplo real de funcionamiento.

La pantalla principal de la aplicación se puede ver en la figura 6.3. donde lo que se puede ver a primera vista es un listado de los DCP conectados a la red Wi-Fi, indicando su nombre y porcentaje de altura en tiempo real. En la barra de herramientas existe un botón de sincronizar, que refrescará el listado, y un menú desplegable que mostrará las opciones de “Añadir controlador” y “Nuevo grupo”. En parte inferior de la barra de herramientas se pueden ver que existen dos ventanas, la que se muestra actualmente “DCP” y la de “Grupos” que se tratará más adelante.

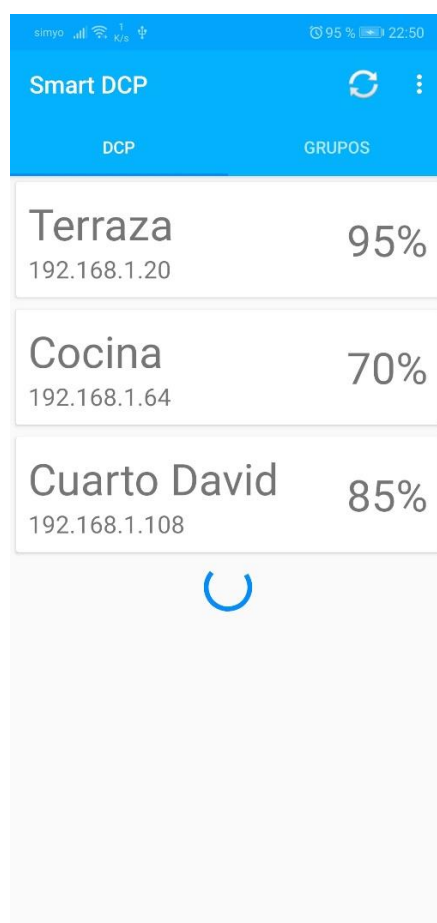


Fig. 6.3. Aplicación - Pantalla principal (DCP). Fuente: Elaboración propia

En el ejemplo de uso actual se muestran en el listado de dispositivos un total de tres persianas: “Terraza”, “Cocina” y “Cuarto David”. Siguiendo los pasos de la figura 6.4. podemos añadir un nuevo dispositivo a la red Wi-Fi. En primer lugar, pulsar “Añadir

controlador”, aparecerá una ventana emergente donde se mostrará la red Wi-Fi a la que está conectada el smartphone y se deberá ingresar la contraseña. Tras darle al botón de “Buscar”, se deberá esperar unos segundos y si se detecta el nuevo DCP correctamente aparecerá en la pantalla principal.

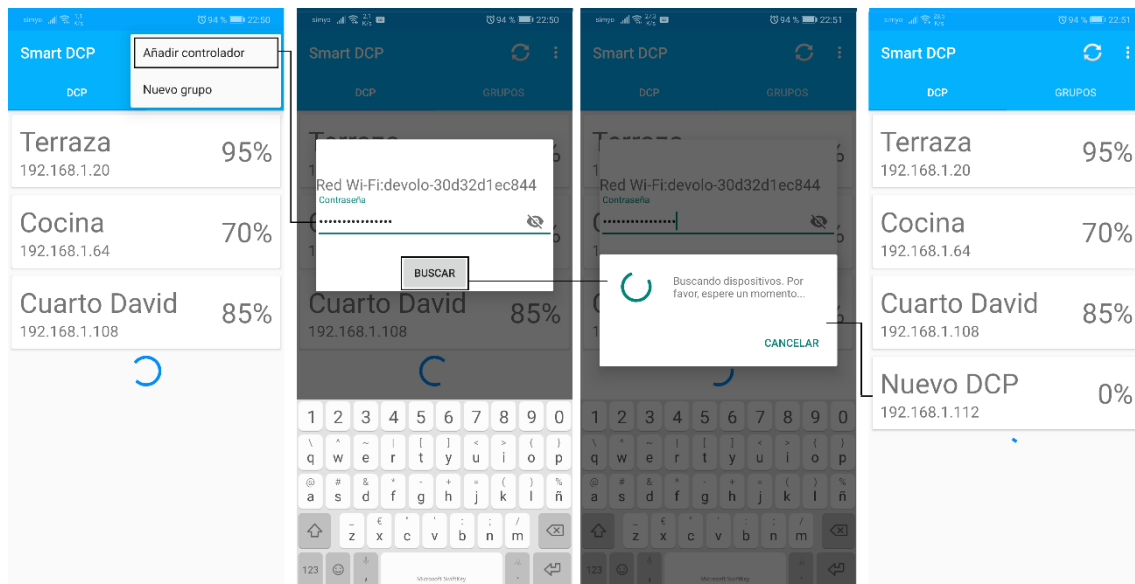


Fig. 6.4. Aplicación – Añadir un DCP a la red Wi-Fi. Fuente: Elaboración propia

Al seleccionar uno de los dispositivos mostrados en el listado de la pantalla principal, se accederá a su configuración. En la figura 6.5. podemos ver a la izquierda la pantalla principal de configuración, que pone a disposición del usuario las distintas opciones de configuración posibles. Los elementos más importantes son la barra de progreso situada debajo del nombre del dispositivo, con la cual se puede ajustar la altura de la persiana a la deseada, y la programación actual del dispositivo. Utilizando el botón de “más” o pulsando en cualquier elemento de la programación horaria actual, aparecerá una ventana emergente con la cual modificar los movimientos automáticos de la persiana. Para confirmar los cambios de la programación horaria se debe pulsar el botón con el icono de guardar.

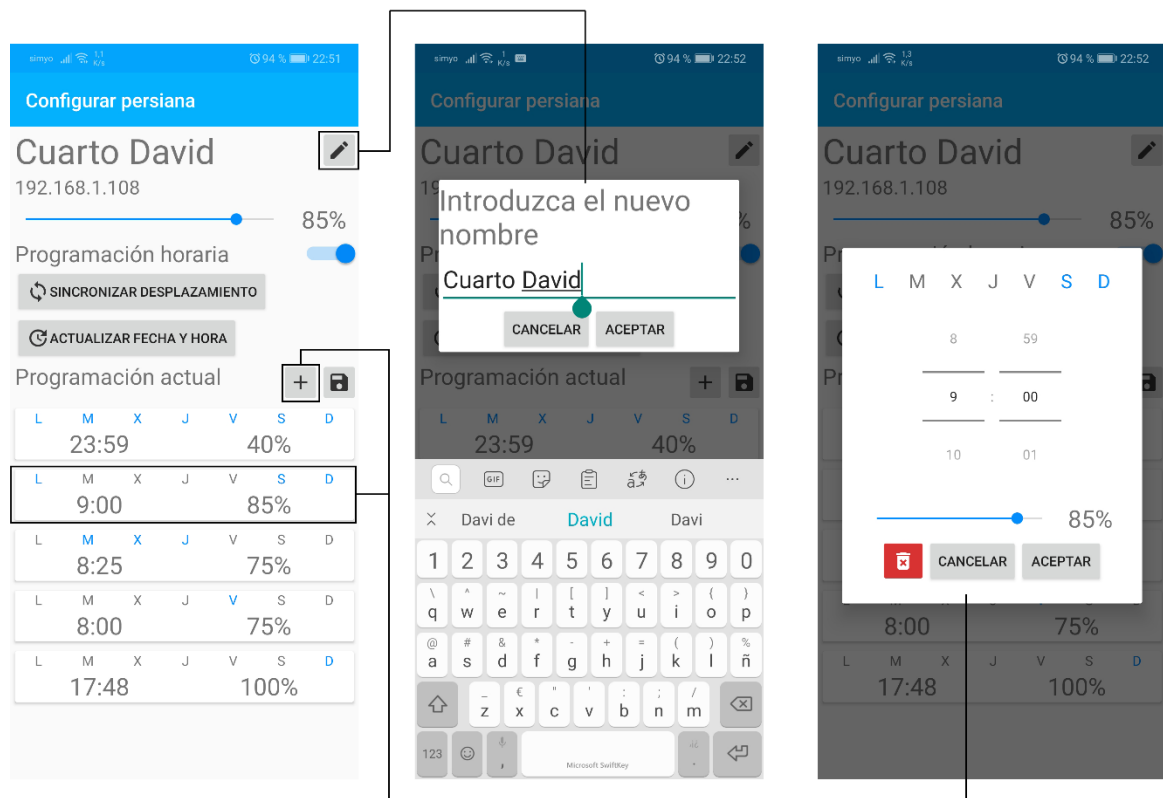


Fig. 6.5. Aplicación – Configuración de un DCP. Fuente: Elaboración propia

Volviendo a las opciones de la pantalla principal, haciendo click en “Nuevo grupo”, nos aparecerá una ventana emergente para ingresar el nombre deseado para el nuevo grupo. Al crear un nuevo grupo se podrá visualizar en la pantalla principal, en la ventana de “Grupos”.

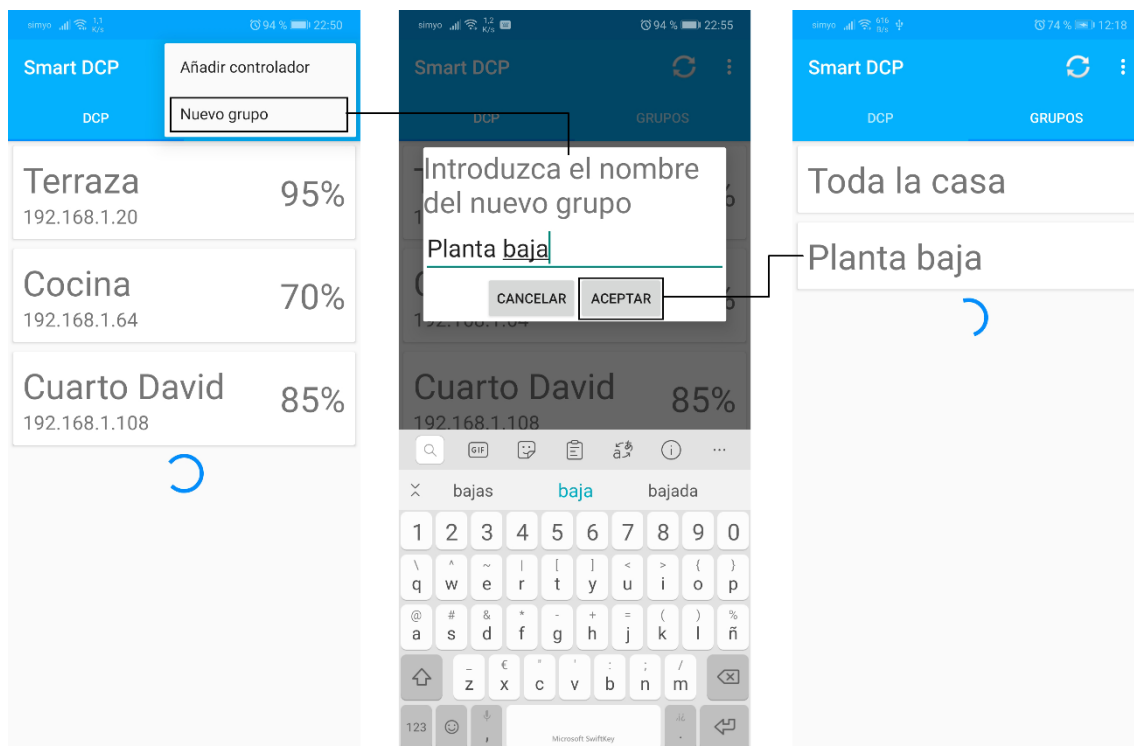


Fig. 6.6. Aplicación – Crear un nuevo grupo. Fuente: Elaboración propia

Al igual que con los DCP, también se puede seleccionar un grupo de la lista y acceder a su configuración. En la figura 6.7. se muestra la pantalla principal de configuración del grupo “Planta baja” al cual se le han añadido las persianas “Cocina” y “Terraza”. Las opciones principales que ofrece la configuración de un grupo son muy similares a las de un DCP individual, permitiendo controlar todas las persianas a la vez con la barra de progreso o mediante la programación horaria del grupo. La principal diferencia reside en la sección de dispositivos asociados al grupo, donde se muestran todos los dispositivos que ya pertenecen a este grupo, y si pulsamos en el botón “más” nos mostrará un listado con el resto de los dispositivos existentes en la red Wi-Fi pudiendo seleccionar el deseado y añadirlo al grupo. Al añadir un nuevo dispositivo al grupo se guardará automáticamente la programación horaria en él. Por último, se dispone de un botón de “borrar” en la barra de herramientas el cual permite eliminar el grupo, borrando automáticamente la configuración del grupo de todos sus dispositivos asociados.

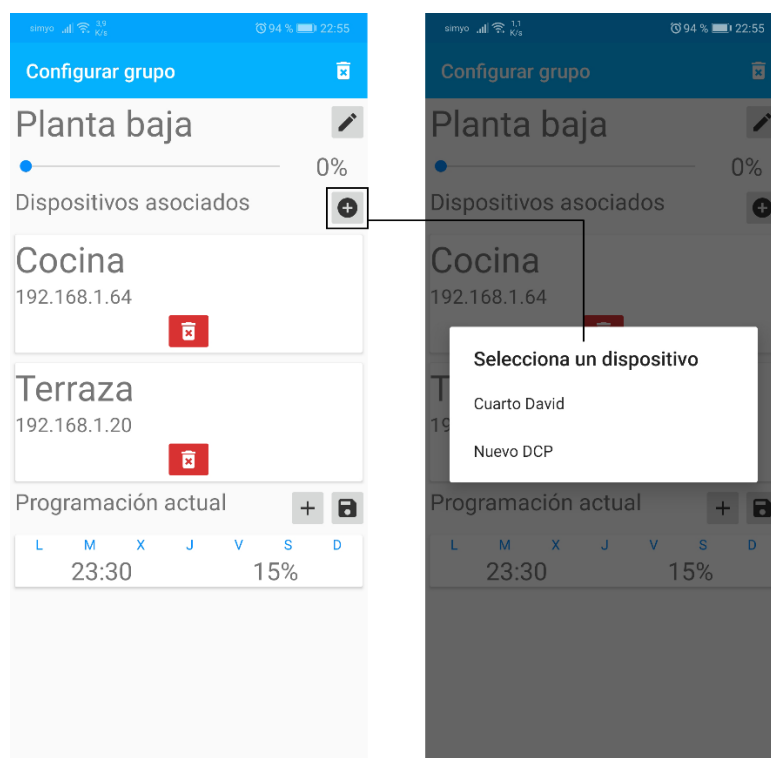


Fig. 6.7. Aplicación – Configuración de un grupo. Fuente: Elaboración propia

6.3. Resultado general del proyecto.

Como resultado del proyecto se ha obtenido un dispositivo domótico instalable en cualquier persiana con el cableado estándar e incluso otro tipo de dispositivos que cumplan el mismo esquema eléctrico, como podría ser un toldo o puerta motorizada. El dispositivo domótico permite un control manual básico, sin embargo, la aplicación ofrece un control a distancia desde cualquier lugar de la misma red Wi-Fi, así como visualizar el estado de cada persiana en tiempo real y la opción de automatizar los movimientos de las persianas a las horas deseadas.

Teniendo esto en cuenta, se puede afirmar que la combinación entre el DCP y la aplicación Android ha permitido alcanzar satisfactoriamente los objetivos definidos para el proyecto, así como todos los requisitos de funcionalidad definidos.

Además, el proyecto pretendía alcanzar los objetivos establecidos buscando un precio final bajo del producto domótico. En la tabla 6.1. se muestra un desglose del coste de los componentes del DCP, resultando en un precio final de 14,29€ (**catorce euros con veintinueve céntimos de euro**). Este es el precio de coste de los componentes adquiridos en España durante el desarrollo del proyecto.

TABLA 6.1. COSTE UNITARIO DEL DCP

Concepto	Coste (incluido impuestos)
ESP32	7,60€
Módulo relé de dos canales	2,09€
Módulo RTC DS3231	2,60€
Piezas auxiliares (cables, botones, leds y resistencias)	2,00€
Coste total	14,29€

Fuente: Elaboración propia

Si bien es un precio considerablemente inferior a lo que ofrece el mercado por un producto con funcionalidad similar, el coste se podría reducir aún más adquiriendo los componentes en el extranjero, aproximando un coste unitario de 5€ (**cinco euros**).

7. PLANIFICACIÓN Y COSTES

En esta sección del documento se realiza una planificación inicial del proyecto, así como la estimación de costes que supondrá llevarlo a cabo. Además, se realizará un contraste de la planificación con los tiempos realmente dedicados al proyecto y de los costes de este. Por último, se estudiará el impacto socioeconómico que podría suponer la aplicación del producto obtenido en este proyecto.

7.1. Planificación inicial.

La realización del presente documento se llevará en paralelo junto a las diferentes fases del proyecto. Durante la evolución y desarrollo del proyecto se plasmará en la documentación cualquier tipo de información, análisis, decisiones de diseño y problemas referentes a la fase que se esté tratando, reflejándose en la planificación esta dedicación de tiempo. El documento incluye ciertas secciones que no se relacionan con ninguna fase del proyecto, la dedicación temporal requerida se imputará en la última tarea de la planificación, llamada “Creación del documento”.

En la planificación inicial se estiman 124 días para el desarrollo del proyecto, que comienzan con una fase de investigación, con una duración de 16 días. En primer lugar, se investigarán los principales microcontroladores que ofrece el mercado, y las capacidades de estos, así como estudios sobre proyectos relacionados con los que tener un primer enfoque para afrontar el proyecto. Además, se manipularán brevemente microcontroladores en este periodo para familiarizarse con su funcionamiento y conseguir una primera experiencia en la programación de estos dispositivos, facilitando y restando trabajo en la fase de diseño y desarrollo. Durante 14 días se estudiarán y definirán formalmente todas las capacidades y funcionalidades del proyecto, estimando los próximos 56 días para el diseño de solución e implementación del DCP y la aplicación móvil. Se aproximan 18 días para probar y validar por completo el correcto funcionamiento del proyecto, incluyendo margen suficiente para corregir pequeños errores que se puedan detectar durante esta fase. Finalmente se estiman 20 días adicionales para completar toda la documentación del proyecto, aunque parte del contenido se ha llevado en paralelo al desarrollo del proyecto.

TABLA 7.1. PLANIFICACIÓN INICIAL.

Tarea	Días	Fecha de inicio	Fecha de fin
TFG	124	25/01/2020	27/05/2020
Investigación	16	25/01/2020	09/02/2020
Investigación microcontroladores	9	25/01/2020	02/02/2020
Investigación proyectos de domótica	7	03/02/2020	09/02/2020
Análisis	14	10/02/2020	23/02/2020
Estudio de funcionalidad del proyecto	11	10/02/2020	20/02/2020
Definición de requisitos	3	21/02/2020	23/02/2020
Diseño y desarrollo	56	24/02/2020	19/04/2020
Diseño y desarrollo del DCP	28	24/02/2020	22/03/2020
Diseño y desarrollo de la aplicación	28	23/03/2020	19/04/2020
Pruebas	18	20/04/2020	07/05/2020
Pruebas del DCP	4	20/04/2020	23/04/2020
Pruebas de la aplicación	5	24/04/2020	28/04/2020
Pruebas del funcionamiento conjunto	9	29/04/2020	07/05/2020
Creación del documento	124	25/01/2020	27/05/2020

Fuente: Elaboración propia

7.1.1. Diagrama de Gantt.

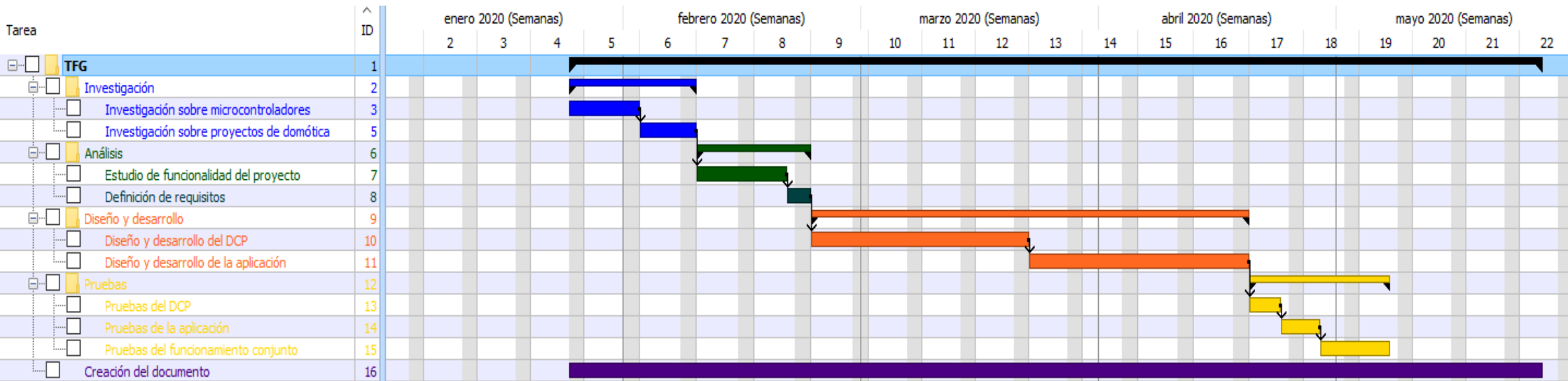


Fig. 7.1. Diagrama de Gantt de la planificación inicial. Fuente: Elaboración propia.

7.2. Estimación de costes.

En esta sección del documento se pretende reflejar el coste estimado para la realización del proyecto y posteriormente se comparará con coste real tras su realización.

El proyecto ha sido planificado para una duración de 124 días, suponiendo la implicación de un único trabajador y una carga de trabajo de 3 horas diarias (fines de semana incluidos). Esto implica un total de 372 horas de trabajo y asumiendo un salario bruto de 14€ por hora, resulta en un coste total de 5.208€ en el sueldo de trabajador durante todo el proyecto (impuestos incluidos). A esto hay que añadir el tiempo y coste de las reuniones con el supervisor del proyecto, estimadas en 20 horas y un coste de 21€ por hora, dejando un total de 420€ en recursos humanos.

Para la realización del proyecto será necesaria la compra de diferentes microcontroladores y componentes eléctricos y electrónicos, así como un motor de persiana eléctrica con el que poder validar el funcionamiento del producto.

TABLA 7.2. ESTIMACIÓN DE COSTES – HARDWARE.

Concepto	Coste (incluido impuestos)
Microcontroladores	50€
Componentes auxiliares	65€
Motor de persiana eléctrica	80€
Coste total	195€

Fuente: Elaboración propia

Existen otros gastos de carácter mensual que también se deben considerar, como el internet o la luz. Para imputar el coste del ordenador utilizado se pondrá como referencia el modelo “Dell Vostro 15 5568” con un precio de 675€ y suponiendo una vida útil de 5 años, implica un coste mensual de 11,25€. Durante el proyecto se utilizará software de carácter gratuito, por lo que no supondrá un gasto adicional.

TABLA 7.3. ESTIMACIÓN DE COSTES – MENSUAL.

Concepto	Coste mensual	Coste imputado (4,13 meses)
Luz	25€	103,25€
Internet	15€	61,95€
Dell Vostro 15 5568	11,25€	46,46€
Coste total	51,25€	211,66€

Fuente: Elaboración propia

Una vez realizados estos cálculos, se puede estimar el coste total que supondrá la realización del proyecto.

TABLA 7.4. ESTIMACIÓN DE COSTES – TOTAL

Concepto	Coste imputado (incluido impuestos)
Sueldo del trabajador	5.208€
Recursos humanos	420€
Microcontroladores	50€
Componentes auxiliares	65€
Motor de persiana eléctrica	80€
Luz	103,25€
Internet	61,95€
Dell Vostro 15 5568	46,46€
Coste total del proyecto	6.034,66€

Fuente: Elaboración propia

7.3. Presupuesto.

A continuación, se presenta el presupuesto que se le ofrecería a un cliente, calculado a partir de los costes del proyecto y añadiendo un riesgo y beneficio al proyecto.

Se ha establecido un porcentaje de riesgo del proyecto en un 10%, de acuerdo con los valores estudiados en la carrera. Además, este valor de riesgo podrá amortiguar pérdidas, debido a que, por la duración del proyecto, un pequeño error en la planificación estimada podría suponer un gran impacto en el coste del proyecto, que principalmente reside en el sueldo del trabajador.

Los beneficios del proyecto se originan en la venta del producto desarrollado, pero al realizar un presupuesto para un cliente, se añadirá un 25% como porcentaje de beneficios.

TABLA 7.5. PRESUPUESTO.

Concepto	Coste imputado (incluido impuestos)
Gastos de personal	5.628€
Gastos de hardware	195€
Otros gastos	211,66€
Total gastos (sin riesgo)	6.034,66€
Riesgo (10%)	603,47€
Total gastos (sin beneficio)	6.638,13€
Beneficios (25%)	1.659,53€
Total presupuesto	8.297,66€

Fuente: Elaboración propia

El presupuesto total del desarrollo del proyecto asciende a 8.297,66€ (**ocho mil doscientos noventa y siete euros con sesenta y seis céntimos de euro**).

Por último, dado que este TFG es un proyecto de desarrollo, para el cálculo de costes no se ha tenido en cuenta la gestión de las tareas de mantenimiento, despliegue, etc., ya que esas tareas se consideran posteriores a la entrega del trabajo.

7.4. Dedicación real.

A continuación se tratan las dedicaciones reales de tiempo y el contraste con la planificación inicial realizada calculando el desvío de los días invertidos en cada tarea.

El proyecto finalmente ha durado 148 días, suponiendo una diferencia de 28 días más de lo planificado inicialmente. Este incremento del tiempo se debe principalmente a tres factores. El primero de ellos ha sido la dificultad para dedicar las horas estimadas en la planificación, ya que, debido a otras tareas tanto universitarias como externas a la universidad, algunas semanas no se ha podido dedicar el tiempo planeado. En segundo lugar, la aparición de dificultades durante la fase de implementación ha supuesto un gran aumento de los días dedicados a esta tarea. Por último, se debe tener en cuenta la situación vivida este año 2020 con la enfermedad COVID-19, lo que ha conllevado un aplazamiento de las fechas de entrega de los TFG dando más margen a los estudiantes.

TABLA 7.6. DEDICACIÓN REAL.

Tarea	Días	Fecha de inicio	Fecha de fin	Desvío respecto al inicial
TFG	148	25/01/2020	20/06/2020	+19%
Investigación	14	25/01/2020	07/02/2020	-13%
Investigación microcontroladores	10	25/01/2020	03/02/2020	+11%
Investigación proyectos de domótica	4	04/02/2020	07/02/2020	-43%
Análisis	11	08/02/2020	18/02/2020	-21%
Estudio de funcionalidad del proyecto	7	08/02/2020	14/02/2020	-36%
Definición de requisitos	4	15/02/2020	18/02/2020	+33%
Diseño y desarrollo	97	19/02/2020	25/05/2020	+73%
Diseño y desarrollo del DCP	52	19/02/2020	10/04/2020	+86%
Diseño y desarrollo de la aplicación	45	11/04/2020	25/05/2020	+61%
Pruebas	4	26/05/2020	29/05/2020	-78%
Pruebas del DCP	1	26/05/2020	26/05/2020	-75%
Pruebas de la aplicación	1	27/05/2020	27/05/2020	-80%
Pruebas del funcionamiento conjunto	2	28/05/2020	29/05/2020	-78%
Creación del documento	22	30/05/2020	20/06/2020	+10%

Fuente: Elaboración propia

Como se puede ver en la tabla 7.6. la dedicación real difiere bastante de la planificada inicialmente. En la mayoría de los casos simplemente se trata de unos pocos días más o menos respecto a lo estimado, pero donde realmente hay una diferencia significativa es en la fase de “Diseño y desarrollo” y “Pruebas”. La fase de “Diseño y desarrollo” se ha visto principalmente afectada por que los tiempo de implementación han sido mayores a lo esperado, sin embargo, el hecho de realizar pruebas constantemente durante el

desarrollo del DCP como de la aplicación ha permitido reducir enormemente el periodo final de validación “Pruebas”, el cual ha llevado tan solo 4 días.

7.5. Costes reales.

En esta sección se calcularán los costes reales del proyecto y se compararán con la estimación realizada al inicio.

La duración final del proyecto ha sido de 148 días, aunque debido a motivos explicados en la sección 7.4. “Dedicación real” la dedicación diaria se ajusta a 2 horas respecto de las 3 pensadas inicialmente. Estos datos nos dejan un total de 296 horas dedicadas al proyecto que, manteniendo el sueldo de 14€ brutos por hora, nos deja un total de 4.144€.

En la tabla 7.7. se desglosa el coste de los componentes hardware necesarios para la implementación y pruebas durante todo el proyecto.

TABLA 7.7. COSTE REAL – HARDWARE.

Concepto	Coste (incluido impuestos)
Microcontroladores	48€
Componentes auxiliares	47€
Motor de persiana eléctrica	50€
Coste total	145€

Fuente: Elaboración propia

Respecto a los costes mensuales, se mantiene la estimación del coste por mes mostrada en la tabla 7.3. de 51,25€. Debido al aumento de la duración del proyecto, el nuevo coste imputado correspondiente a 4,93 meses ascendiendo la cifra hasta los 252,66€.

Haciendo el cálculo total del coste del proyecto vemos un total de 4.961,66€, que son 1.073€ menos de lo estimado inicialmente. El principal motivo de esta diferencia se debe a que, a pesar de que la duración del proyecto ha sido mayor de la estimada, el cómputo global de horas dedicadas ha sido menor.

TABLA 7.8. COSTE REAL – TOTAL.

Concepto	Coste imputado (incluido impuestos)
Sueldo del trabajador	4.144€
Recursos humanos	420€
Microcontroladores	48€
Componentes auxiliares	47€
Motor de persiana eléctrica	50€
Luz	123,25€
Internet	73,95€
Dell Vostro 15 5568	55,46€
Coste total del proyecto	4.961,66€

Fuente: Elaboración propia

Suponiendo la venta del proyecto con el presupuesto inicial de 8.297,66€ y los costes finales de 4.961,66€ la diferencia nos deja con 3.336€ (**tres mil trescientos treinta y seis euros**) de beneficio bruto.

7.6. Impacto socioeconómico.

La incorporación del producto desarrollado en este proyecto en el mercado presentaría ventajas tanto en el contexto social como en el económico.

Desde el punto de vista social, el producto en cuestión permite la automatización y control de una o varias persianas de manera sencilla, mejorando la calidad de vida de las personas que utilizan el producto.

En el aspecto económico podemos distinguir dos beneficios. En primer lugar, la venta comercial del producto supondría una actividad lucrativa para la empresa que se encargue de la fabricación y distribución, así como la creación de puestos de trabajo. En segundo lugar, supone un beneficio para los consumidores, ofreciendo un producto que permite la domotización de persianas de una forma mucho más económica y ampliando la accesibilidad para adquirir este tipo de productos. Los principales perjudicados serían las empresas involucradas en este sector, por la aparición de una competencia más barata, lo que les podría suponer un decremento de su actividad económica y/o sus márgenes de beneficio.

8. CONCLUSIONES Y TRABAJOS FUTUROS

8.1. Conclusiones.

Con este proyecto se quería conseguir un sistema completamente funcional de control y automatización de persianas, capaz de competir en utilidad y precio con lo que ofrece el mercado. El proyecto se conforma por el desarrollo de un componente hardware capaz de controlar un motor de persiana común y su implementación software que permitiese una comunicación y control a través de una red Wi-Fi, así como la configuración de desplazamientos automáticos por día y hora. Esta idea requería el desarrollo de una aplicación móvil desde la que controlar y configurar todas las persianas domotizadas a través de la conexión Wi-Fi.

El correcto estudio y diseño del dispositivo de control de persianas (DCP), ha permitido alcanzar un nivel de funcionalidad que hace frente a la competencia actual en el mercado. El dispositivo permite el control de la persiana mediante botones integrados en el circuito y ofrece feedback al usuario sobre el modo de funcionamiento y el estado de la configuración Wi-Fi a través de diodos led. Se ha utilizado un módulo relé que permite controlar el paso de corriente de alto voltaje, con el que cual bajar y subir la persiana eléctrica. La incorporación de un módulo de reloj y el almacenamiento en memoria secundaria de ciertas variables del software implementado le permiten una completa recuperación tras cortes en el suministro eléctrico. Todo esto es gestionado por un microcontrolador, siendo el ESP32 el seleccionado para este proyecto. Además, el desarrollo de la aplicación móvil permite controlar y gestionar todos los dispositivos de la red Wi-Fi de manera sencilla, incluso pudiendo crear grupos de dispositivos para poder gestionar varios al mismo tiempo. El almacenamiento distribuido de la configuración en los dispositivos de control permite que los usuario con acceso a la red Wi-Fi y la aplicación instalada puedan gestionar las persianas, pudiendo acceder a la configuración establecida por otro usuario. Una selección precisa de los componentes ha permitido alcanzar toda la funcionalidad con un coste a partir de los 5€ frente a la competencia con un coste que superior a los 20€.

El trabajo resultante ha cumplido satisfactoriamente los objetivos perseguidos, ofreciendo un producto domótico que permite controlar y automatizar todas las persianas de una casa a muy bajo coste con el único requerimiento de una red Wi-Fi como método de comunicación entre los dispositivos y el smartphone, permitiendo una gestión remota desde cualquier lugar de la red local. Además, se debe tener en cuenta que se ha conseguido el desarrollo del proyecto con un coste inferior al presupuesto inicial.

Uno de los aspectos diferenciadores de este proyecto respecto los productos que existen actualmente en el mercado es que el dispositivo de control creado es capaz de conocer la altura a la que se encuentra la persiana y conseguido sin ningún tipo de sensor adicional. Como se explica a lo largo del proyecto, esto se ha logrado gracias a una

sincronización inicial donde se mide el tiempo de subida y bajada de la persiana, y realizando un cálculo de altura cada vez que se desplaza la persiana. Tras ver el buen resultado que aporta esta solución y su simplicidad sin incrementar el coste en hardware, se podría plantear su uso en dispositivos comercializados.

8.2. Líneas futuras de trabajo.

Durante las fases de análisis y diseño de este proyecto han surgido diferentes alternativas sobre las que se ha tenido que elegir, así como la aparición de ideas que se decidió no llevar a cabo. A continuación, se exponen las principales líneas futuras del proyecto.

La incorporación de sensores de luminosidad en las ventanas permitiría condicionar la altura de la persiana en función de la cantidad de luz solar que se recibe. Esto permitiría al usuario establecer un umbral con la máxima y mínima cantidad de luz que quiere recibir y el microcontrolador se encargaría de hacer los desplazamientos de manera automática para ajustarse los valores establecidos.

Similar al caso anterior, la adición de un sensor de temperatura podría utilizarse para controlar la temperatura de la vivienda, bajando las persianas en las horas de más calor de manera automática. La incorporación de esta funcionalidad al sistema no solo mejora la experiencia del usuario si no que puede suponer un ahorro energético reduciendo el uso del A/C entre otros.

Se podría estudiar utilizar el hardware diseñado y adaptar el software implementado para un uso específico en puertas de garaje. En este caso, la apertura y cierre mediante una programación horaria no cobraría mucha importancia, salvo para casos muy específicos, pero la utilización de sensores de proximidad o la localización del dispositivo móvil podría ser de utilizad para automatizar el movimiento de la puerta de garaje.

Un avance en la funcionalidad del sistema se conseguiría utilizando un servidor externo, permitiendo la conexión entre los controladores y el dispositivo móvil cuando no se encuentran en la misma red Wi-Fi. De esta forma, el usuario podría ver como se encuentran las persianas, por ejemplo, de su casa, sin estar en ella. Esto permitiría el control del sistema desde cualquier sitio con acceso a internet.

9. ENGLISH COMPETENCE

9.1. Introduction.

The growing trend of domotics combined and move by the emergence of smartphones, and more recently, by the visual assistants, has increased the number of household items that can be controlled and automated. The technological evolutions produced this decade have opened a lot of possibilities, allowing the domotization and connectivity of many elements such as lights, thermostats and smart plugs.

This work was born from the personal idea of making a fully functional home automation project on which to apply and reflect all the knowledge acquired. This project tries to design and develop a hardware component capable of being coupled to a motorized blind system, as well as the software that will allow the control and programming of the blind from a mobile device. The main purpose is to achieve a device suitable for the general public and with similar or superior capacities and characteristics to those currently offered on the market. In addition, should be a low-cost device, lowering the overall price that can involve the domotization of a home, opening this possibility to a greater number of people.

Although this project is based and oriented to the domotization of electric blinds, the resulting product will be perfectly usable in other elements that follow the same scheme of electric motor, such as a garage door.

9.2. Objectives.

This project follows the achievement of three main objectives:

- **O1:** The first objective of the project is the design and creation of a hardware device based on a microcontroller, adding the necessary components to allow the control of the motor of an electric blind. In addition, the blind control device (abbreviated as DCP) must allow a connection and communication via Wi-Fi.
- **O2:** The second objective is the software implementation of the DCP, handling requests via Wi-Fi to control the motorized blind. Also, it must be able to store a schedule that allows the blind to move autonomously at the times set by the user.
- **O3:** As a third objective, the development of an application for mobile devices capable of managing, controlling and displaying the status of the blinds connected through the home automation system. In addition, it must allow the configuration of the schedule of the automatic movements of the blinds.

9.3. Market research.

At this time the market offers a variety of possibilities when it comes to automating and controlling electric blinds, each with its advantages and disadvantages. Below are some of the most important options.

Somfy is an experienced company highly involved in the research and development of technologies applicable to smart homes. They have a wide variety of control and automation devices for heating, lighting, garage doors and, of course, blinds and awnings. In addition, they have an exclusive application for each type of product they sell, as well as physical controls to manage them. The products offered by Somfy are of high quality, although this leads to an excessively high price, especially when compared to their market alternatives. However, their main disadvantage is the compatibility of their products, since their mobile controls and applications are only compatible with "io-homecontrol" devices. Therefore, if you want to automate an already installed electric blind, it would not be possible, having to replace the complete electric blind kit with one of the brand. [10]

We can find other alternatives in the market such as those offered by the company Maxcio. The product offered consists of a blind switch, modified to be used by voice commands from Alexa and Google Assistant, with the option of a time schedule to automate the system. Its connectivity is established via Wi-Fi and is intended for local area networks (WLAN). As strong points, it stands out for its compatibility with most of the electric blind motors, as well as its low cost, which is around 30€. [11]

9.4. Project scope.

The project can be easily differentiated in two parts, the one referred to the functionality, design and implementation of the DCP and its corresponding in the mobile application.

The application will allow users to control the domotic system of blinds of the Wi-Fi network in which they are connected. DCPs can be managed, allowing the creation of groups to control several blinds in one action, avoiding having to control one by one. There will be the option of automating a blind or group by establishing a time schedule for the DCPs. The mobile application will be available for the Android operating system.

The DCPs will be able to control the electric motor of the blind, being able not only to raise/lower the blind completely, but also to establish intermediate positions. The DCP must allow a manual control of the blind, and giving the user the option to temporarily disable the scheduled moves. There will be LEDs on the DCP that will provide feedback to the user of the device status.

9.5. Design.

The project have tow components that must be designed, the blind control device (DCP) and the mobile APP. Both components will communicate through a common Wi-Fi network where the blind controller will send the current status information over the network and will handle control and configuration requests sent by the smartphone.

9.5.1. Wi-Fi communications.

In this system we can find three different types of communications between the APP and the DCP.

First, a data transmission with the DCP as the sender and the mobile devices running the application as the target receiver. The UDP protocol will be used for the packet transmission over the network using the broadcast technique. The use of broadcast is necessary for the mobile devices to identify the DCPs for the first time, because they will not know their IP addresses at the first time.

The second type of communication will be based on the execution of requests by the mobile application to the DCPs, which will act as servers. This case will be the one that will perform the main information flow between the two modules of the architecture and the one that will allow the control of the blind from the mobile device. For this purpose, a TCP protocol will be used, which offers a reliable communication in a simple way, establishing a connection between the mobile device and the DCP until the data flow required by the application request is completed.

Finally, when you want to add a new DCP, you will have to send it the credentials of the Wi-Fi network, because without them it will not be able to establish the rest of the communications. To carry out this data transmission, the DCP will open a Wi-Fi access point to where the mobile device will be connected and will transmit the credentials in a secure way, encrypting the data.

9.5.2. DCP.

The blind control device (DCP) is the physical component that will control and automate the motorized blind. The design of the component requires a study of the hardware components and the software implementation to achieve the established operating requirements.

The main component of the DCP will be a microcontroller, which will allow the handling of the auxiliary components of the project. There are many microcontrollers focused on IoT projects that meet the requirements of the project. After a study of the main alternatives, has been chosen the use of the ESP32. This is a microcontroller designed by the company Espressif Systems and very common in IoT project developments. Its main features include integrated Wi-Fi and Bluetooth connectivity, a dual-core processor and hardware-accelerated cryptography.

The microcontroller is capable of providing up to 5V to power small external components in DC. Common blind motors are usually powered by the household outlet, in the case of Spain, 230V AC at 50Hz. The microcontroller must be able to control the electric motor, so the best option in this case is to use a relay, which, controlled from the microcontroller, can manage the current passage of an electric circuit at 230V AC.

A real time clock or RTC is an electronic component capable of keeping the current time. Thanks to its low consumption and the existence of an external power supply (such as a button cell) it is tolerant of power cuts. The DCP allows for the establishment of a time schedule with which to automate the movement of the blind, so it is necessary for the microcontroller to know the current time. The microcontroller is capable of maintaining the current time while it is executing, but if a power failure occurs, the microcontroller will lose the time value. By using an RTC module, the microcontroller will be able to recover the current date and time value and continue normal operation after power is restored.

The complete circuit design is shown in figure 5.2.

9.5.3. APP.

There are currently two main mobile markets to launch an application. Firstly, iPhones using the IOS operating system and devices with the Android operating system.

It has been decided to deploy the mobile application on the Android operating system, in the Android Studio development environment and with the Java programming language. The main reason is the existing knowledge of the development team with the Android environment which will facilitate the implementation and performance testing.

The prototyping of the user interface gives an initial idea of the visual result of the project, in which the necessary elements of the application are quickly and easily reflected. The creation of Mock-Ups of the main screens of the APP has allowed to obtain a prototype of user interface with a low temporary cost that reflects in a simple way that functionality he information that is wanted to implement in the final application.

The main screen (shown in figure 4.4.) should show the list of all connected DCPs and the groups that have been created. The configuration of each device can be accessed intuitively by selecting the corresponding element from the list, which will take you to the control screen (shown in figure 4.5.). In addition, there is a toolbar with the refresh and add buttons. The refresh button will reload all the information of the APP, while with the add button you can create groups or add a new DCP.

The control screen (shown in figure 4.5.) shows the configuration and status of a DCP device or group. First, the name associated with the DCP or group is displayed, in this example it is the device associated with the living room blind. Next to the name there is an edit button that will allow you to modify it. The next element is a progress bar, which will allow us to modify the height of the blind by sliding and dropping it at the desired height. The rest of the elements shown correspond to the configuration of the device's time schedule. A small switch allows us to activate or deactivate the complete time programming of the device without having to delete the current configuration. Then, all the configured displacements will be shown as a list. In the example shown there is a single movement that registers a displacement, opening the blind completely (100% height) at 9:00 and which will only take effect on the days highlighted (Monday to Friday). Selecting one of the programmed movements or pressing the add button gives access to the screen for editing the time schedule (shown in figure 4.6), allowing you to modify or create movements.

9.6. Results.

In this section an analysis of the results obtained will be made, explaining and detailing all the implemented functionality and how to use it. A section will be differentiated for the results of the DCP and another one for the application. In addition, a global analysis of the results provided by the project.

9.6.1. DCP results.

The hardware component designed and implemented, called DCP, has been achieved thanks to a study of the necessary components and the development of its software. The circuit shown in figure 6.1. is a real example of implementation of the device in a protoboard.

Once you have the circuit, the blind motor must be connected to the household current and the DCP. The project is designed and defined to be able to control any blind motor with the standard connections, these are: a cable for the grounding, a cable for the neutral connection of the plug and two cables for the phase connection (one to go up and one to go down). The ground and the neutral must be connected in the normal way

to the electrical installation of the home and it will be the two up/down cables that will be connected to the phase by means of the relay module, as shown in figure 6.2.

Once the DCP has been installed, the functionality it offers with the physical control only is limited to its three buttons. The up and down buttons are used to activate the motor, allowing the blind to be moved in both directions while the button is pressed. In addition, either of these two buttons can be combined with the function button so that a complete scrolling of the blind can be performed without holding down the scroll buttons. For example, if the blind is lowered and you want to raise it completely, you could press and hold the raise button until you reach the upper limit, but for convenience, if you press the function button and then click the raise button, the blind automatically moves up. In addition, pressing the function button twice in a row will change the operating mode of the DCP, alternating between active and inactive time programming. Finally, if you press and hold the function button for 3 seconds, the DCP will clear all its settings, removing Wi-Fi credentials, time settings, etc...

The DCP has two LEDs that provide the user with basic information about the status of the device. As shown in figure 6.1. the blue led corresponds to the status of the Wi-Fi connection, when no connection has been established or the configured network has been lost, the led will flash, and when a Wi-Fi connection is correctly established the led will remain lit. The green led corresponds to the DCP operating mode, lighting up when the time programming is activated and going out when it is deactivated.

9.6.2. APP results.

The application has been designed to allow quick configuration and communication with the DCP with a simple interface, based on the prototype created in the design phase.

The main screen of the application can be seen in figure 6.3. where what can be seen at first sight is a list of the DCPs connected to the Wi-Fi network, indicating their name and percentage of height in real time. In the toolbar there is a synchronize button, which will refresh the list, and a drop-down menu that will show the options "Añadir controlador" (add controller) and "Nuevo grupo" (new group). In the lower part of the toolbar you can see that there are two windows, the one that is currently shown "DCP" and the one of "Grupos" (groups).

In the example, a total of three blinds are shown in the device list: "Terraza", "Cocina" and "Cuarto David". Following the steps of figure 6.4. we can add a new device to the Wi-Fi network. First, press "Añadir controlador" (add controller), a pop-up window will appear where the Wi-Fi network to which the smartphone is connected will be shown and the password will have to be entered. After pressing the "Buscar" (search) button,

you will have to wait a few seconds and if the new DCP is detected correctly it will appear on the main screen.

By selecting one of the devices shown in the list on the main screen, its configuration will be accessed. Figure 6.5. shows on the left the main configuration screen, which provides the user with the different possible configuration options. The most important elements are the progress bar located under the name of the device, with which the height of the blind can be adjusted to the desired one, and the current programming of the device. Using the "+" button or clicking on any element of the current time schedule, a pop-up window will appear with which you can modify the automatic movements of the blind. To confirm changes to the time schedule, press the button with the save icon.

Returning to the main screen options, clicking on "Nuevo grupo" (new group), a pop-up window will appear to enter the desired name for the new group. When creating a new group, it will be displayed on the main screen, in the groups window.

As with DCPs, you can also select a group from the list and access its settings. Figure 6.7. shows the main configuration screen of the "Planta baja" group to which the "Cocina" and "Terraza" blinds have been added. The main options offered by the configuration of a group are very similar to those of an individual DCP, allowing you to control all the blinds at the same time with the progress bar or by means of the group's time schedule. The main difference lies in the section of devices associated to the group, where all the devices that already belong to this group are displayed, and if we click on the "+" button it will show us a list with the rest of the existing devices in the Wi-Fi network, being able to select the desired one and add it to the group. When adding a new device to the group, the time schedule will be automatically saved in it. Finally, there is a delete button on the toolbar which allows you to delete the group, automatically deleting the group's configuration from all its associated devices.

9.6.3. General results of the Project.

As a result of the project, a domotic device has been obtained that can be installed in any blind with the standard wiring and even other types of devices that comply with the same electrical scheme, such as an awning or motorized door. The domotic device allows a basic manual control, however, the application offers a remote control from anywhere in the same Wi-Fi network, as well as displaying the status of each blind in real time and the option to automate the movements of the blinds at the desired time.

With this in mind, it can be stated that the combination between the DCP and the Android application has allowed to successfully achieve the objectives defined for the project, as well as all the defined functionality requirements.

In addition, the project aimed to achieve the established objectives by seeking a low final price for the domotic product, resulting in a final price of 14,29€ per unit (**fourteen euros and twenty nine cents**). Although it is a considerably lower price than what the market offers for a product with similar functionality, the cost could be further reduced by purchasing the components abroad, approximating a unit cost of € 5 (**five euros**).

9.7. Planning and costs.

In the initial planning, 124 days are estimated for the development of the project, which begins with a research phase, lasting 16 days. First, the main microcontrollers offered by the market and their capabilities will be investigated, as well as studies on related projects to have a first approach to the project. In addition, microcontrollers will be briefly manipulated during this period to become familiar with their operation and get a first experience in the programming of these devices, facilitating and subtracting work in the design and development phase. During 14 days all the capacities and functionalities of the project will be studied and formally defined, estimating the next 56 days for the solution design and implementation of the DCP and the mobile application. 18 days are estimated to test and fully validate the correct operation of the project, including sufficient margin to correct small errors that may be detected during this phase. Finally, 20 additional days are estimated to complete all the project documentation, although some of the content has been taken in parallel to the development of the project. The figure 7.1. shows the initial Gantt chart of the Project.

The project has finally taken 148 days, which is 28 days longer than initially planned. This increase in time is mainly due to three factors. The first one has been the difficulty to dedicate the hours estimated in the planning, since, due to other tasks, some weeks it has not been possible to dedicate the planned time. Secondly, the appearance of difficulties during the implementation phase has meant a large increase in the days dedicated to this task. Finally, the situation experienced this year 2020 with the COVID-19 virus must be taken into account, which has led to a postponement of the TFG initial calendar.

The following is the budget that would be offered to a client, calculated from the project costs and adding a risk and benefit to the project. A project risk percentage of 10% has been established, in accordance with the values studied. In addition, this risk value may cushion losses, because, for the duration of the project, a small error in the estimated planning could have a large impact on the cost of the project, which mainly resides in the employee's salary. The benefits of the project originate from the sale of the developed product, but when a budget is made for a client, 25% will be added as a percentage of profits. The total budget for the development of the project amounts to 8.297,66€ (**eight thousand two hundred and ninety seven euros and sixty six cents**).

The total cost of the project is 4.961,66€, which is 1.073€ less than initially estimated. The main reason for this difference is that, despite the fact that the duration of the project has been longer than estimated, the overall calculation of hours spent has been lower. Assuming the sale of the project with the initial budget of 8.297,66€ and the final costs of 4.961,66€, the difference leaves us with 3.336€ (**three thousand three hundred and thirty-six euros**) of gross profit.

9.8. Conclusions.

With this project we wanted to achieve a fully functional system of control and automation of blinds, capable of competing in utility and price with what the market offers. The project consists of the development of a hardware component capable of controlling a common blind motor and its software implementation that allows communication and control through a Wi-Fi network, as well as the configuration of automatic movements by day and time. This idea required the development of a mobile application from which to control and configure all the domotic blinds through the Wi-Fi connection.

The correct study and design of the blind control device (DCP), has allowed to reach a level of functionality that faces the current competition in the market. The device allows the control of the blind with buttons integrated in the circuit and offers feedback to the user about the operation mode and the state of the Wi-Fi configuration through LEDs. A relay module has been used to control the passage of high-voltage current, with which to lower and raise the electric blind. The incorporation of a clock module and the storage in secondary memory of certain variables of the implemented software allows a complete recovery after power cuts. All this is managed by a microcontroller, being the ESP32 the one selected for this project. In addition, the development of the mobile application allows to control and manage all the devices of the Wi-Fi network in a simple way, even being able to create groups of devices to manage several at the same time. The distributed storage of the configuration in the control devices allows users with access to the Wi-Fi network and the application installed to manage the blinds and being able to access the configuration established by another user. A precise selection of the components has allowed to achieve all the functionality with a cost starting from 5€ against the competition with a cost that exceeds 20€.

The resulting work has satisfactorily achieved the proposed objectives, offering a home automation product that allows to control and automate all the blinds of a house. All of this with a very low cost with the only requirement of a Wi-Fi network as a method of communication between the devices and the smartphone. In addition, it must be taken into account that the development of the project has been achieved at a lower cost than the initial budget.

One of the differentiating aspects of this project with respect to the products currently on the market is that the control device created is capable of knowing at what height the blind is located and achieved without any additional sensor. As explained throughout the project, this has been achieved thanks to an initial synchronization where the time for raising and lowering the blind is measured, and by making a height calculation every time the blind is moved. After seeing the good result that this solution provides and its simplicity without increasing the cost in hardware, one could consider its use in commercialized devices.

9.9. Future work.

To conclude, some alternatives and ideas will be presented as future work for this project.

The incorporation of brightness sensors in the windows would make it possible to condition the height of the blind according to the amount of sunlight received. This would allow the user to establish a threshold with the maximum and minimum amount of light he wants to receive and the microcontroller would automatically make the movements to adjust the established values.

Similar to the previous case, the addition of a temperature sensor could be used to control the temperature of the house, automatically lowering the blinds during the hottest hours. The addition of this functionality to the system not only improves the user experience but can also lead to energy savings by reducing the use of A/C.

Could be studied using the designed hardware and adapting the implemented software for a specific use in garage doors. In this case, the opening and closing by a schedule would not be very important, except for very specific cases, but the use of proximity sensors or the location of the mobile device could be used to automate the movement of the garage door.

An improve in system functionality would be achieved by using an external server, allowing the connection between the controller devices and the mobile device when they are not on the same Wi-Fi network. In this way, the user could see how the blinds are located, for example, at home, without being there. This would allow control of the system from anywhere with internet access.

BIBLIOGRAFÍA

- [1] «Constitución Española,» 29 12 1978. [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-1978-31229>.
- [2] «Boletín Oficial del Estado,» 12 04 1996. [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-1996-8930>.
- [3] «Boletín Oficial del Estado,» 01 04 2010. [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2010-5292>.
- [4] «Boletín Oficial del Estado,» 14 12 1999. [En línea]. Available: <https://boe.es/buscar/act.php?id=BOE-A-1999-23750>.
- [5] «Asociación Española de Domótica e Inmótica,» [En línea]. Available: <http://www.cedom.es/sobre-domotica/que-es-domotica>.
- [6] J. Sabaté, 2013. [En línea]. Available: <https://www.fm-house.com/wp-content/uploads/2014/12/I-Congreso-Edificios-Inteligentes.pdf#page=245>.
- [7] R. A. Mainato Quichimbo, «Universidad del Azuay,» 2011. [En línea]. Available: <http://201.159.222.99/handle/datos/311>.
- [8] P. Turmero, «Monografias.com,» [En línea]. Available: <https://www.monografias.com/trabajos106/conceptos-del-hogar-digital/conceptos-del-hogar-digital2.shtml>.
- [9] A. G. Mendoza Varela, «Universidad Israel,» 2019. [En línea]. Available: <http://repositorio.uisrael.edu.ec/handle/47000/1928>.
- [10] «Somfy,» 2019. [En línea]. Available: <https://www.somfy.es/productos/persianas-y-estores-motorizados/persianas-enrollables>.
- [11] Maxcio, «Maxcio,» 2019. [En línea]. Available: <https://maxcio.com/collections/smart-curtain-switch-1/products/alexa-rolladen-zeitschaltuhr-maxcio-smart-jalousien-vorhang-schalter-kompatibel-mit-alexa-und-google-home-app-fernbedienung-und-timing-funktion-touch-schalter-nullleiter-erforderlich-auf-und-a>.
- [12] Espressif Systems, «Espressif,» 09 01 2020. [En línea]. Available: <https://www.espressif.com/en/products/hardware/esp32/resources>.
- [13] Espressif Systems, «Espressif,» 12 12 2019. [En línea]. Available:

<https://www.espressif.com/en/products/hardware/esp8266ex/resources>.

- [14] Maxim Integrated Products, 2015. [En línea]. Available: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>.
- [15] «Statcounter,» 05 2020. [En línea]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201905-202005>.
- [16] Arduino, «Arduino Reference,» 2020. [En línea]. Available: <https://www.arduino.cc/reference/en/>.
- [17] Espressif, «GitHub - Espressif,» 2020. [En línea]. Available: <https://github.com/espressif/arduino-esp32>.
- [18] «Fritzing Forum,» Enero 2018. [En línea]. Available: <https://forum.fritzing.org/t/esp32s-hiletgo-dev-board-with-pinout-template/5357>.

ANEXO A: MANUAL DE USO

En esta sección se detalla un manual de uso del sistema domótico. Se recoge toda la funcionalidad ofrecida por el producto desarrollado, detallando paso a paso como utilizarla.

1. Instalación del producto.

El primer paso para la domotización de una persiana es la instalación del dispositivo de control. Para ello, se debe identificar cual será la persiana que se quiere domotizar. El dispositivo de control permite controlar y automatizar una persiana motorizada, por lo que la persiana que se quiera domotizar ya debe contar con un kit de motor eléctrico instalado.

Una vez seleccionada la persiana a domotizar, se deben realizar las siguientes conexiones eléctricas entre la persiana y el dispositivo de control:

- Conectar el cable de subida del motor eléctrico (consultar las instrucciones del kit de motor de persiana utilizado) con la conexión “NO1” del relé del dispositivo de control.
- Conectar el cable de bajada del motor eléctrico (consultar las instrucciones del kit de motor de persiana utilizado) con la conexión “NO2” del relé del dispositivo de control.

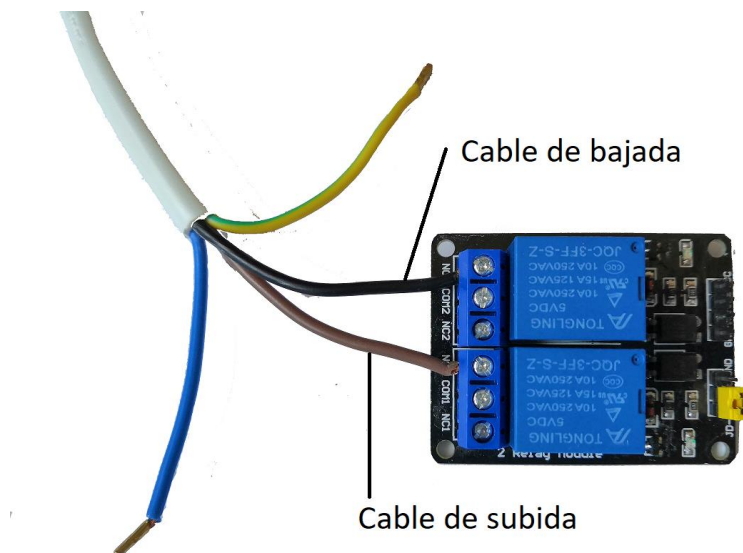


Fig. A.1. Conexiones bajada y subida. Fuente: Elaboración propia

- de subir, la persiana se desplazará hasta el tope superior sin necesidad de mantener pulsado.
- Botón de bajada: Mientras se mantenga pulsado el botón de bajada la persiana se desplazará hacia abajo. Pulsando el botón de función y seguidamente el botón de bajar, la persiana se desplazará hasta el tope inferior sin necesidad de mantener pulsado.
 - Botón de función: Pulsando dos veces seguidas el botón de función se cambiará el modo de funcionamiento, alternando entre programación horaria activada y desactivada. Al mantener pulsado el botón de función durante 3 segundos el dispositivo restaurará los valores por defecto, borrando toda configuración almacenada.

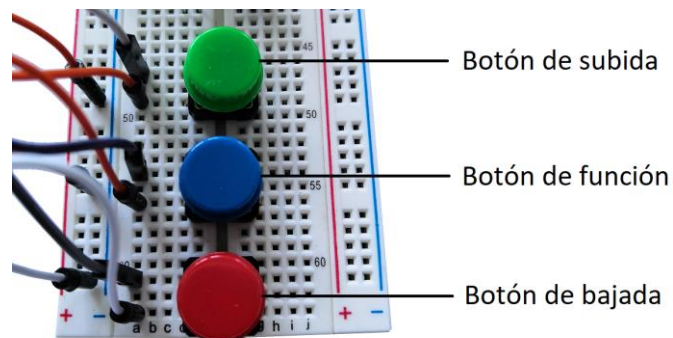


Fig. A.3. Botones del dispositivo de control. Fuente: Elaboración propia

3. Indicadores LED.

El dispositivo de control posee dos indicadores LED con los que se puede ver rápidamente información sobre el estado del dispositivo. El LED verde indica el modo de funcionamiento del dispositivo, cuando se encuentra apagado significa que el dispositivo no realizará los movimientos automáticos que tiene configurados, y cuando está encendido sí realizará los movimientos automáticos. El LED azul muestra el estado de la conexión Wi-Fi, permaneciendo encendido si la conexión está establecida y parpadeando cuando no lo está.

Programación horaria activada/desactivada

Conexión Wi-Fi

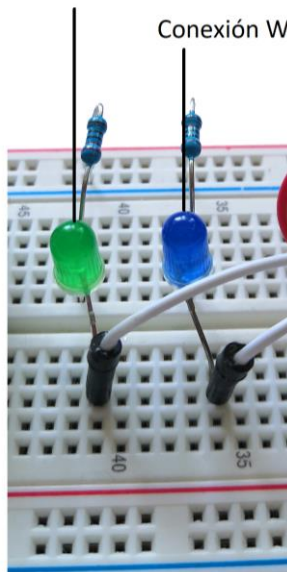


Fig. A.4. Indicadores de estado LED. Fuente: Elaboración propia

4. Configuración de un nuevo dispositivo.

Cuando se instala un nuevo dispositivo se debe realizar una configuración inicial para poder usarlo con normalidad. Esta configuración inicial se realizará desde la aplicación móvil.

En primer lugar, se debe configurar la conexión con la red Wi-Fi. Se debe acceder a la opción “Añadir controlador”, verificar que la red Wi-Fi mostrada es la correcta e introducir la contraseña. Tras darle al botón de buscar y esperar unos segundos se deberá configurar automáticamente la conexión Wi-Fi, pudiendo verificar que el led de estado Wi-Fi del dispositivo permanece encendido.

El siguiente paso que se debe realizar tras la conexión Wi-Fi del dispositivo es realizar una sincronización de la altura de la persiana. El dispositivo utiliza valores por defecto para el tiempo de desplazamiento de la persiana, pero cada motor y persiana son diferentes, por lo que es altamente recomendable realizar este proceso de calibración para que el funcionamiento sea el correcto.

Para llevar a cabo la sincronización de altura se deben seguir los siguientes pasos:

1. Seleccionar desde la aplicación la persiana que se quiere calibrar.
2. Pulsar el botón de “Sincronizar desplazamiento”.
3. Utilizar los botones físicos del dispositivo para bajar por completo la persiana.
4. Pulsar el botón de “Comenzar” en la aplicación.
5. Mantener pulsado el botón de subir que aparece en la aplicación mientras se pulsa al mismo tiempo el botón físico de subir la persiana del dispositivo. Cuando la persiana alcance el límite superior soltar ambos botones.

6. Mantener pulsado el botón de bajar que aparece en la aplicación mientras se pulsa al mismo tiempo el botón físico de bajar la persiana del dispositivo. Cuando la persiana alcance el límite inferior soltar ambos botones.

Tras realizar estos pasos la persiana quedará calibrada. Este proceso se puede repetir en cualquier momento, por lo que no hay ningún problema si se realiza algún error durante el proceso de sincronización o si la persiana no queda calibrada correctamente.

Por último, es posible que la hora almacenada en el dispositivo no coincida con la actual. En cualquier momento se puede utilizar el botón de “Actualizar fecha y hora” para actualizar estos valores en el dispositivo.

5. Gestión de dispositivos.

La aplicación móvil permite un control y configuración de las persianas domotizadas en la propia red Wi-Fi.

En la pantalla inicial de la aplicación existen dos pestañas diferentes. En la primera podemos ver un listado de las persianas conectadas actualmente y desplazando lateralmente accedemos a la pestaña con los grupos existentes.

Seleccionando una persiana accedemos a su pantalla de control y configuración. Los dispositivos recién instalados aparecerán con el nombre de “Nuevo DCP”, utilizando la opción de editar nombre se podrá modificar en cualquier momento. La barra de progreso se utiliza para mover la persiana a una nueva posición, siendo 0% totalmente bajada y 100% totalmente subida, permitiendo también valores intermedios. La programación horaria se puede activar y desactivar utilizando el interruptor habilitado para ello. Los botones de “Sincronizar desplazamiento” y “Actualizar fecha y hora” se han explicado detalladamente en el punto anterior. La programación horaria se muestra en forma de listado, con todos los desplazamientos configurados para la persiana seleccionada. Para cada desplazamiento se muestra los días de la semana que se llevará a cabo y la hora, así como la nueva posición que debe tomar. Para modificar un desplazamiento bastará con pulsar en él y modificar los parámetros, y se podrán añadir nuevos desplazamientos utilizando el botón de “más”. Por último, cuando se realicen cambios en la programación horaria se debe guardar utilizando el botón correspondiente.



Fig. A.5. Opciones de control de una persiana. Fuente: Elaboración propia

Seleccionando un grupo existente también se accede a su configuración, que es muy parecida a cada persiana individualmente. El nombre del grupo que se podrá modificar utilizando el botón de editar. La barra de progreso y la programación horaria funcionan de igual manera que para una persiana individual, pero en este caso se aplicarán los cambios en todas la persianas pertenecientes al grupo. La principal diferencia es la sección de “Dispositivos asociados” que muestra un listado con las persianas pertenecientes al grupo. Utilizando el botón de “más” se podrán añadir persianas al grupo y con el botón de papelera eliminar cualquier persiana del grupo.

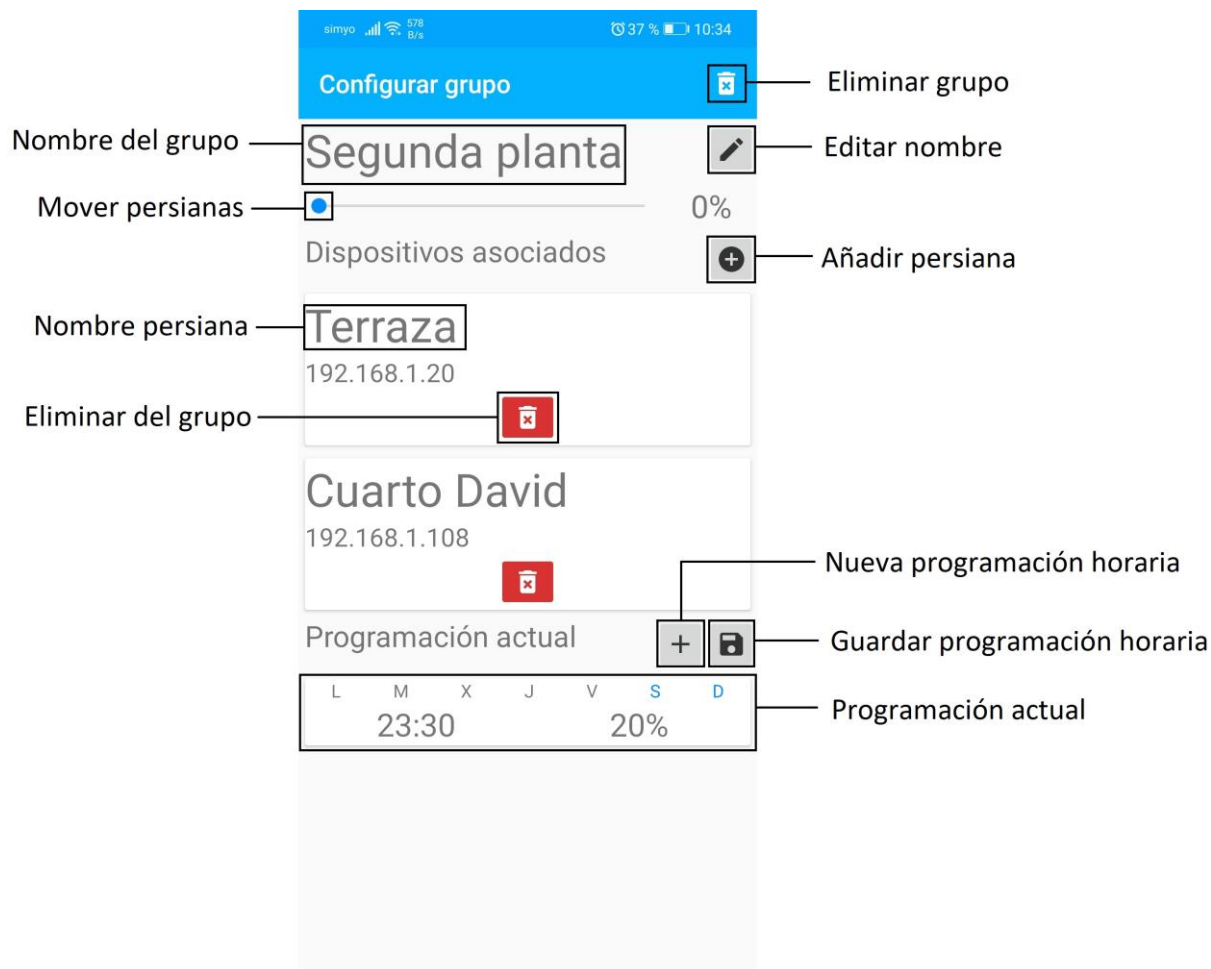


Fig. A.6. Opciones de control de un grupo. Fuente: Elaboración propia